4/84

DM 6,—
öS 52,—
sfr 6,—
hfl 6,80

c't magazin für computer technik

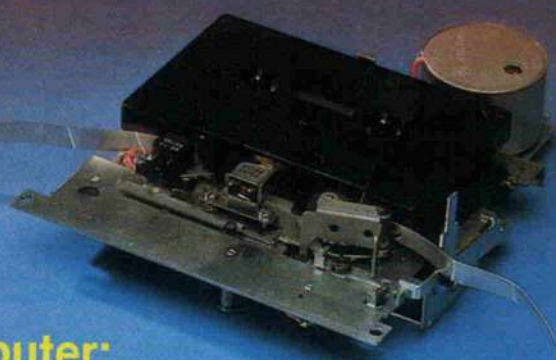
c't-Serie
BASIC intern
★ Was nicht im Handbuch steht ★

Was heißt CP/M-kompatibel?

BASIC-Benchmarks

Casio FP-200

MAX 1



Für jeden Computer:

Speichern, Lesen, Kommunizieren mit 3600/7200 Baud

Centronics-Schnittstellen für ZX 81, VC-20, C 64

c't — ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 27 46, 3000 Hannover 1

4 April 1984

Heise Software-Service

Ein Extra-Service des Heise-Software-Service

VICTAPE COMPUTING
für VC-20

SPECTRUM COMPUTING
für ZX-Spectrum

MODEL B COMPUTING
für BBC-Acorn, Modell B

Spezial-Magazine auf Compact-Kassetten

COMPUTING demonstriert, was Ihr Homecomputer kann:

★ Systemprogramme ★ Spiele
★ Bewegliche Grafik ★ Internationale News

auf dem Farbbildschirm

COMPUTING-Magazine — exklusiv für c't-Leser
in der internationalen Originalausgabe (in englischer Sprache)

Alle zwei Monate

Preis: 19,80 DM (zuzüglich 3 DM Versandkosten)

Für Sinclair-Fans:

ZX-COMPUTING

das große Spezialmagazin für ZX81 und Spectrum
(Originalausgabe in englischer Sprache)

Nummer 4

Nummer 5

Programme und Informationen
über Ihren Computer

Preis: 9,80 DM (zuzüglich 1,70 DM Versandkosten)

Bestellen beim

Heise-Software-Service
Postfach 27 46 · 3000 Hannover 1

Lieferung nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck oder den quittierten Einzahlungsbeleg Ihrer Bank bei. Überweisungen bitte auf das Konto-Nr. 93 05-308 Postscheckamt Hannover

c't-Abonnement

Abrufkarte

GARANTIE

Wir garantieren jedem Abonnenten das Recht, seine Bestellung innerhalb einer Woche nach Abschluß schriftlich zu widerrufen.

c't-Abonnement Abrufkarte

Ja, übersenden Sie mir bis auf Widerruf alle künftigen c't-Ausgaben ab Monat

_____

(Kündigung 8 Wochen zum Jahresende möglich.)

Das Jahresabonnement kostet DM 58,— inkl. Versandkosten und MwSt.

Absender und Lieferanschrift
Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Straße/Nr.

PLZ Wohnort

_____

Datum/Unterschrift

Ich bestätige ausdrücklich, vom Recht des schriftlichen Widerrufs innerhalb einer Woche nach Abschluß beim Verlag Heinz Heise GmbH, Postfach 27 46, 3000 Hannover 1, Kenntnis genommen zu haben.

Unterschrift _____
Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

c't-Kontaktkarte

Mit dieser Service-Karte können Sie

- **Informationen** zu in **c't** besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierenden oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Folien, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, **c't**-Versand, Postfach 27 46, 3000 Hannover 1, **ordern**.

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in **c't** _____/8__, Seite _____ erschienene
☐ Anzeige ☐ redaktionelle Besprechung
☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____
☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't-Kontaktkarte

Mit dieser Service-Karte können Sie

- **Informationen** zu in **c't** besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierenden oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Folien, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, **c't**-Versand, Postfach 27 46, 3000 Hannover 1, **ordern**.

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in **c't** _____/8__, Seite _____ erschienene
☐ Anzeige ☐ redaktionelle Besprechung
☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____
☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't-Abonnement **Abrufkarte**

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleitzahl Konto-Nr.

Geldinstitut Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Antwortkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

magazin für computer technik

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

c't-Abonnement
Abrufkarte

Abgesandt am

_____ 198__

zur Lieferung ab

Heft _____ 198__

Jahresbezug DM 58,—
inkl. Versandkosten und MwSt.

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen. ▶

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

_____ 198__

an Firma _____

Bestellt/angefordert

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen. ▶

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

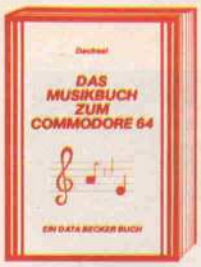
Abgesandt am

_____ 198__

an Firma _____

Bestellt/angefordert

NEUE SUPERHITS RUND UM COMMODORE



Der COMMODORE 64 ist ein Musikgenie, und mit diesem Buch lernen Sie alles über seine musikalischen Fähigkeiten. Der Inhalt reicht von einer Einführung in die Computermusik über die Erklärung der Hardware-Grundlagen und die Programmierung in BASIC bis hin zur fortgeschrittenen Musikprogrammierung. Zahlreiche Beispielprogramme und leicht verständliche Darstellung. Erschließen Sie sich die Welt des Sounds und der Computermusik mit dem MUSIKBUCH ZUM COMMODORE 64.
Ca. 200 Seiten, DM 39,–

Graphik ist eine der Hauptstärken des COMMODORE 64. Mit diesem neuen Buch lernen Sie, wie Sie die graphischen Fähigkeiten programmtechnisch optimal nutzen, von einfachen Figuren über Sprites, Zeichensatzprogrammierung, Hardcopy und IRQ-Handhabung bis hin zu Funktionendarstellung, Statistik, 3-D, CAD und den Geheimnissen der Actionsspiele. Zahlreiche Beispielprogramme ergänzen dieses Buch, das die faszinierende Computergraphik jedermann zugänglich macht.
Ca. 250 Seiten, DM 39,–

Umfassendes Nachschlagewerk zum COMMODORE 64 und seiner Programmierung. Allgemeines Computerlexikon mit Fachwissen von A–Z und Fachwörterbuch mit Übersetzungen wichtiger englischer Fachbegriffe – das DATA BECKER LEXIKON ZUM COMMODORE 64 stellt praktisch drei Bücher in einem dar. Es enthält eine unglaubliche Vielfalt an Informationen und dient so zugleich als kompetentes Nachschlagewerk und als unentbehrliches Arbeitsmittel. Ein Muß für jeden COMMODORE 64 Anwender!
Ca. 350 Seiten, DM 49,–

Nicht nur alles über Interfaces und Ausbaumöglichkeiten des COMMODORE 64 enthält dieses Buch, sondern auch über seine vielfältigen Einsatzmöglichkeiten von der Lichtorgel über Motorsteuerung, Spannungs- und Temperaturmessung bis zur programmierbaren Stromversorgung. Zehn komplette Schaltungen zum Selberbauen, vom Eprommer über den Digital-Voltmeter mit automatischer Messbereichswahl und den Logic Analyzer bis zur preiswerten Spracheingabe-Sprachausgabe. Jeweils mit Schaltplan, Layout und Softwarelisting.
Ca. 220 Seiten, DM 49,– ab April 84.

64 FÜR EINSTEIGER ist eine sehr leicht verständliche Einführung in Handhabung, Einsatz, Ausbaumöglichkeiten und Programmierung des COMMODORE 64, die keinerlei Vorkenntnisse voraussetzt. Schritt für Schritt führt das Buch Sie in die Programmiersprache BASIC ein, wobei Sie nach und nach eine komplette Adressenverwaltung erstellen, die Sie anschließend nutzen können. Das Buch ist sowohl als Einführung als auch als Orientierung vor dem 64er Kauf gut geeignet.
Ca. 200 Seiten, DM 29,–

Diese neue, umfangreiche Programmsammlung hat es in sich. Über 50 Spitzenprogramme für den COMMODORE 64 aus den unterschiedlichsten Bereichen, vom Superspiel über Graphik- und Soundprogramme sowie Utilities bis hin zu Anwendungsprogrammen. Der Hit sind zu jedem Programm aktuelle Programmiertips und Tricks der einzelnen Autoren zum Selbermachen. Also – nicht nur abtippen, sondern auch dabei lernen und wichtige Anregungen für die eigene Programmierung sammeln.
Ca. 250 Seiten, DM 49,–

... und natürlich die bewährten Bestseller aus bester Hand

Insgesamt über 200 000mal wurden die nachfolgenden Bücher in nur 12 Monaten verkauft. Machen auch Sie mehr als Ihrem COMMODORE mit diesen beliebten und bewährten Bestsellern aus bester Hand.

Endlich ein umfangreiches Trainingshandbuch, das Ihnen detailliert den Umgang mit SIMON's BASIC erklärt. Ausführliche Darstellung aller Befehle und ihrer Anwendung. Zahlreiche Beispielprogramme und Programmiertricks. Dieses Buch sollte jeder SIMON's BASIC Anwender haben!
ca. 300 S., DM 49,–

Eine leicht verständliche Einführung in das Programmieren des C-64 in Maschinensprache und Assembler. Komplett mit vielen Beispielen sowie einem Assembler, Disassembler und einem Einzelschritt-Simulator. Und natürlich zugeschnitten auf Ihren Computer, den COMMODORE 64.
ca. 200 S., DM 39,–

64 INTERN erklärt detailliert Technik und Betriebssystem des C-64 und die Programmierung von Sound und Graphik. Ausführlich dokumentiertes ROM-Listing, zahlreiche lauffertige Beispielprogramme und 2 Original-Schaltpläne zum Ausklappen. Dieses Buch sollte jeder 64-Anwender und Interessent haben.
ca. 320 S.; DM 69,–

64 TIPS & TRICKS ist eine echte Fundgrube für jeden COMMODORE 64 Anwender. Umfangreiche Sammlung von POKE's und anderen nützlichen Routinen, BASIC-Erweiterungen, Graphik und Farbe für Fortgeschrittene, CP/M, Multitasking, mehr über Anschluß- und Erweiterungsmöglichkeiten und zahlreiche lauffertige Programme.
ca. 290 S.; DM 49,–

64 FÜR PROFIS zeigt, wie man erfolgreich Anwendungsprobleme in BASIC löst und verrät Erfolgsgeheimnisse der Programmierprofis. 5 komplett beschriebene, lauffertige Anwendungsprogramme (z. B. Adreßverwaltung) illustrieren den Inhalt der einzelnen Kapitel beispielhaft. Mit diesem Buch lernen Sie gute und erfolgreiche BASIC-Programmierung.
ca. 320 S., DM 49,–

DAS GROSSE FLOPPY-BUCH erklärt detailliert die Arbeit mit der Floppy VC-1541, von der sequentiellen Datenspeicherung bis zum Direktzugriff, für Anfänger, Fortgeschrittene und Profis. Ausführlich dokumentiertes DOS-Listing, zahlreiche lauffertige Beispiel- und Hilfsprogramme, z. B. Disk Editor und Haushaltsbuchführung.
ca. 320 S.; DM 49,–

VC-20 INTERN ist für jeden interessant, der sich näher mit Technik und Maschinenprogrammierung des VC-20 auseinandersetzen möchte. Detaillierte technische Beschreibung des VC-20, ausführliches ROM-Listing, Einführung in die Maschinenprogrammierung und 3 Original-Schaltpläne.
ca. 230 S.; DM 49,–

VC-20 TIPS & TRICKS ist eine echte Fundgrube für jeden VC-20 Anwender. Sound und Graphik Programmierung, Speicherbelegung und Speichererweiterung, BASIC-Erweiterungen, POKE's und andere nützliche Routinen, zahlreiche lauffertige Beispiel- und Anwendungsprogramme und vieles andere mehr.
ca. 230 S.; DM 49,–

Darauf haben Sie gewartet!

Die neue DATA WELT ist eine Computerzeitschrift speziell für COMMODORE-Anwender. Brandaktuell (detaillierte Informationen über die neuen COMMODORE Computer 264 und 364) und randvoll mit Berichten, Trends und interessanten Programmiertips. 80 Seiten stark im Magazinformat. Gleichzeitig als Nachfolger des VC-Infos umfassende Übersicht über aktuelle Produkte, Bücher und Programme rund um COMMODORE 64 und VC-20. Die Frühjahrsausgabe der neuen DATA WELT erhalten Sie ab Anfang März überall dort, wo es DATA BECKER BÜCHER und -PROGRAMME gibt. Am besten gleich holen oder direkt bei DATA BECKER gegen DM 4,– in Briefmarken anfordern.

BESTELL-COUPON c't 4

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
☐ DATA WELT 1/84 (DM 4,– in Briefmarken liegen bei

Name und Adresse bitte deutlich schreiben

IHR GROSSER PARTNER FÜR KLEINE COMPUTER

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010 · im Hause AUTO BECKER

DATA BECKER BÜCHER und PROGRAMME erhalten Sie im Computer-Fachhandel, in den Computerabteilungen der Kauf- und Warenhäuser und im Buchhandel. Auslieferung für Österreich Fachbuch-Center ERB, Schweiz THALI AG und Benelux COMPUTERCOLLECTIEF.

Inhaltsve

c't 4/84

8 **Leserbriefe**

16 **Software-Service**

107 **Platinen-Service**

12 **c't-aktuell**

102 **c't-Buchkritik**

107 **Inserenten-Verzeichnis**

108 **Vorschau auf Heft 5/84**

108 **Impressum**

c't-Prüfstand

29 **Langsam, aber gründlich**
Hand-held für unter 1000 DM: Casio FP 200

32 **MAX aus Deutschland**
Vielseitiger Kleincomputer

Software-Know-how

42 **Grafik, die auf jedem Rechner läuft**
Balken und Kurven in Microsoft-BASIC

45 **BASIC intern**
Was nicht im Handbuch steht

62 **Benchmark-Test**
— eine Computer-Meßlatte?

64 **FORTH anpassen**
Teil 2: Kommunikation mit FORTH

68 **GENIE als ASCII-Fernschreiber**
Serieller Datenverkehr im Eintonverfahren

72 **Was heißt CP/M-kompatibel?**

85 **Arithmetik für 6502 und Z 80**
Teil 2: Multiplizieren und Dividieren

c't-Titel

SuperTape

Vor den Programm- und Datenaustausch zwischen Computern verschiedenen Fabrikats haben die Hersteller viele Hindernisse gestellt. Eine der größten Hürden besteht im Einsatz unterschiedlicher Formate und Verfahren für die Aufzeichnung auf Datenträgern. c't will dieses Problem aus der Welt schaffen. Wir stellen ein schnelles und sicheres Kassettenformat vor, das voraussichtlich für alle populären Homecomputer verwendet werden kann. SuperTape ermöglicht eine Übertragungsrate von (wahlweise) 3600/7200 Baud. Auf eine C60-Kassette passen also bis zu 3,2 MByte. Selbstverständlich eignet sich das Verfahren auch für die direkte Datenübertragung von Rechner zu Rechner. Übrigens: Für das c't-Projekt des Monats brauchen Sie diesmal keinen Lötkolben.

Seite 21

SuperTape mit dem Z 80

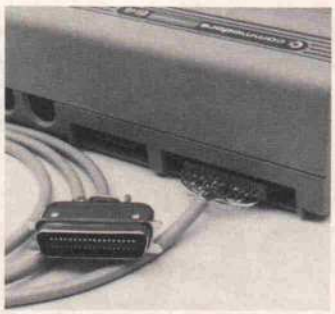
SuperTape-Basisroutinen für Z 80 zum Senden und Empfangen von Daten — dargestellt am Beispiel des ZX 81 — finden Sie ab

Seite 23

Centronics-Schnittstellen für VC-20, C 64, ZX 81

Die meistverbreiteten Homecomputer besitzen keine Schnittstellen zum Anschluß von Standard-Druckern. c't schließt diese Lücke mit einer Serie von Beiträgen. In dieser Ausgabe finden Sie Software-Lösungen für VC-20 und C 64 und ein sehr preisgünstiges Interface für den ZX 81 nebst Treiberprogramm für Centronics-kompatible Drucker.

Seite 92



Casio FP 200 auf dem Prüfstand

Ein Hand-Held-Computer für weniger als 1000 Mark. Kann der Casio neben seinen deutlich teureren Konkurrenten bestehen? Der c't-Test offenbart Stärken und Schwächen des FP 200.

Seite 29

Was heißt CP/M-kompatibel?

CP/M-Software läuft auf jedem CP/M-Rechner — diese Aussage gilt für fast alle kommerziell angebotenen Programme. Probleme tauchen aber dann auf, wenn man Software selbst schreibt. Dieser Bericht zeigt, wann ein Programm als kompatibel gelten darf und welche Bedingungen eingehalten werden müssen.

Seite 74

erzeichnis

MAX aus Deutschland

Der Hinweis 'Made in Germany' ist sicher überflüssig bei einem Computer mit dem Namen 'MAX'. Mit MAX kann man, laut Prospekt, rechnen, schreiben, kreativ spielen, programmieren, wecken, messen, steuern und regeln. Ein wahrer Tausendsassa aus deutschen Landen also. Näheres im c't-Test auf

Seite 32

Grafik, die auf jedem Rechner läuft

Daß die grafische Darstellung oft aussagekräftiger ist als endlose Zahlenkolonnen, ist eine Binsenweisheit. Doch nicht jeder Rechner ist grafikfähig, und erst recht nicht jeder Drucker. Deshalb wurden in der EDV schon früh Techniken entwickelt, die notfalls sogar die Ausgabe auf einem Fernschreiber gestatten. Die Lösungen in Microsoft-BASIC ab

Seite 42

Benchmark-Test — eine Computer-Meßlatte?

Viele Leser wollten wissen, welche 'Philosophie' hinter den Benchmark-Tests steht, die auf dem c't-Prüfstand ausgeführt werden. In diesem Bericht stellen wir die Programme vor und beleuchten zugleich kritisch den Sinn solcher Geschwindigkeitsvergleiche.

Seite 62

Einfacher A/D-Wandler

Soll ein Computer Vorgänge aus der 'analogen Welt' erfassen, überwachen oder regeln, so müssen diese in die Digitalform übersetzt werden. In diesem Praxistip stellen wir einen einfachen und sehr preisgünstigen Analog/Digital-Wandler vor, der sich zur genauen Umsetzung von Spannungswerten eignet.

Seite 78

Die neue c't-Serie: BASIC intern — Was nicht im Handbuch steht

Wie funktionieren Interpreter und Compiler? Wie kann man das 'serienmäßige' BASIC erweitern? Mit welchen Tricks arbeiten die Programmierer von Microsoft? — Eine c't-Serie für alle, die mehr als die 'Benutzeroberfläche' ihres Computers kennenlernen möchten. Also auch für Assembler-Freaks.

Seite 45

Super-Chips

Ein Mikroprozessor mit Großrechner-Leistung: der NS32032 ist derzeit der leistungsfähigste Baustein aus der 16000-Familie von National, deren Architektur sich von herkömmlichen deutlich unterscheidet. Unseren Report über die Mikro-Avantgarde finden Sie auf

Seite 80

c't-Programm

WordStar als Grafik-Editor
Blockgrafik mit OSBORNE 1 und MX 80 — **75**

Computer zu Hause

Zahlen Sie zuviel Steuern?
Gesamtlohnsteuertabellen 1984 — **49**

Mischkulturen
Spectrum 48 K als Hobbygärtner — **52**

c't-Projekte

SuperTape
Ein Kassettenformat für alle Computer — **21**

SuperTape mit dem Z 80
Am Beispiel des Sinclair ZX 81 — **23**

Das Betriebssystem CP/M 86
Anpassung an den c't 86, Teil 2 — **36**

Nur das Kabel löten
Centronics-Schnittstelle für VC-20 und C 64 — **92**

Centronics-Interface für ZX 81 — **96**

c't-Praxistip

Einfacher A/D-Wandler
CA 3162 in 6502- und Z 80-Systemen — **78**

c't-Report

Super-Chips
Nationals 32032 und seine Geschwister — **80**

Grundlagen

I/O-Techniken
Drucken per Interrupt — **57**

c't erscheint 12mal im Jahr zur Monatsmitte.

c't 5/'84 erhalten Sie ab **12. 4. 84.**

Nichts zu retten

Heute habe ich Ihre 3. Ausgabe von c't in Händen und kann schon nach kurzer Durchsicht sagen, daß es auch hier nichts mehr zu retten gibt! Es freut mich, daß ich diese Zeitschrift nicht gleich abonniert habe, obgleich ich das nach den ersten 2 Ausgaben tun wollte!

Anscheinend gibt es aber auch unter Ihren 'Schreibern' VC-20/C-64- und Sinclair-'Freaks', die es nicht lassen können, weiteren 'Sabbel' darüber zu verbreiten! Wenn ich die Computer-Zeitschriften so anschaue, kommt es mir vor, als gäbe es nur 2 Computer — den Commodore und Sinclair!?! Nicht, daß ich verärgert bin, daß man keinen Beitrag über Genie bringt — darauf kann ich auch verzichten, aber 'Listing-Hefte' gibt es doch schon genug — oder? In Ihre Zeitschrift gehören m. E. nur Beiträge, wie z. B. der Bau des c't-86, des Terminals, Floppy usw., wie man es in den 'anderen' nicht mehr findet! Wenn Sie davon abweichen, werden Sie viele Leser verlieren — als ersten mich!

Martin Schuster, Heidenheim

Mehr Grundsätzliches

Ich möchte Ihre Zeitschrift bisher insgesamt loben. Insbesondere fand ich positiv, daß Sie bei Konstruktionsentwürfen immer einen 'Grundsatzartikel' zu einem der eventuell unbekannteren Bauteile brachten (6511Q beim Terminal, 8086 beim c't 86) und möchte Sie bitten, damit nicht etwa aufzuhören, sondern dies eher noch auszuweiten. So wäre ich an einer genaueren Beschreibung des SC-01A interessiert gewesen.

Sie verweisen immer wieder auf die Datenblätter usw. der Hersteller (u. a. bei der Beschreibung des 6511Q). Leider konnte ich jedoch in den Elektronikläden der Umgebung solche Datenblätter nicht erhalten und auch keine Auskunft darüber, wer sie hier in Deutschland vertreibt. Wäre es Ihnen möglich, dies in Zukunft immer mit zu veröffentlichen?

C. Flothow, Sinsheim

Ihre Anregungen nehmen wir uns zu Herzen. Datenblätter erhalten Sie beim Hersteller oder dessen Repräsentanten. Hier die Anschrift der deutschen Rockwell-Vertretung:

Rockwell International GmbH
Fraunhoferstr. 11
8033 München-Martinsried
(Red.)

c't-Terminal-Computer: Nur eine Werbebeilage

Um es vorwegzunehmen, Ihr o. g. Artikel erscheint mir — etwas überspitzt formuliert — als Werbebeilage der Hardware-Lieferfirma.

Mein Hauptkritikpunkt: Der Artikel ist im Anspruchsniveau entschieden zu hoch, jedenfalls nur schwer nachvollziehbar. Letzteres scheint mir beabsichtigt.

Sie haben recht damit, daß nicht jeder Artikel bei 'Adam und Eva' beginnen kann. Doch ist beispielsweise die praktische Umsetzung von Blockschaltbild 4 (c't 12/83, S. 44) in die Schaltung schlecht beschrieben. Außerdem bin ich der Meinung, daß ein Terminal für 'den anspruchsvollen Anwender' mit einer deutschen Tastatur beginnt. Die Angaben zum Anschluß einer solchen sind aber sehr mangelhaft. Die Aussage, daß ein Assembler-Listing nicht vorliegt, weil es 'bislang keinen Assembler für den ... 6511-Prozessor' gibt, ist schlicht eine Ausrede: Sie hätten die Abkürzungen aus Tab. 1 (c't 1/84, S. 89) verwenden können und so mindestens exemplarische Teile des Programms nachvollziehbar machen können. Erst so sind die von Ihnen häufig angesprochenen 'Programmanpassungen' möglich.

Der Entwickler bzw. der Artikelschreiber hat sich wahrscheinlich wochenlang mit der speziellen Materie beschäftigt. Er muß diese aber auch Leuten, die nicht hauptberuflich mit solchen Entwicklungen befaßt sind, vermitteln können — so jedenfalls verstehe ich das Konzept einer Zeitschrift wie der Ihren.

Allerdings drängt sich mir der Verdacht auf, daß dies gar nicht beabsichtigt ist. Zwar ist es durchaus legitim, den Gewinn durch das Zusammenspiel zwischen Zeitschrift und Hardware-Entwicklern (bzw. Lieferfirmen) zu maximieren, wenn dabei der Lerneffekt jedoch völlig auf der Strecke bleibt, kann ich auf den Kauf der Zeitschrift verzichten und gleich einen fertigen Computer kaufen.

Ich hoffe, Sie verstehen meine Kritik als Anregung.

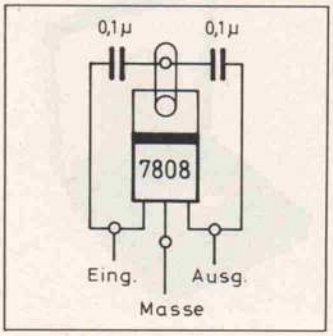
Helmut Bartz, Frankfurt

Da sind Sie auf dem Holzweg, ein Komplott zur Verschleierung von Know-how gibt es nicht. Es liegt wohl eher in der Natur komplexer Projekte, daß sie höhere Anforderungen an das Verständnis stellen. Dem Terminal-Projekt hat c't bisher zwölf Druckseiten mit — zugegeben — konzentrierter Information gewidmet. Dies halten wir für einen tragbaren Kompromiß zwischen den Interessen der Leser, die das Terminal bauen und eventuell nach eigenen Vorstellungen verändern möchten, und denjenigen, denen andere Themen wichtiger sind. (Red.)

Den Kopf halt kühl

Ein dickes Lob an Fa. Sinclair! Sie stattete ihren Spectrum anwenderfreundlich mit einer aufwendigen, thermostatisch gesteuerten Heizungsanlage aus. Als Heizelement dient ein 5-V-Spannungsregler-IC, der so bei 90°C, abschaltet. Allerdings ergibt sich ein kleiner Punkt der Kritik:

Die CPU wird hierdurch auch erwärmt. Damit wird aber nicht folgenlos gegen die Regel '... den Kopf halt kühl ...' verstoßen. Kurzum der Spectrum wird nach einiger Zeit eigenwillig (Variablen mit mehreren Buchstaben lassen sich nicht mehr eingeben). Deshalb ist es unerläßlich, die 15V des Netzteils, wie bereits für den ZX81 empfohlen, durch ein Spannungsregler-IC 7808 auf 8V herabzusetzen. Als Nebenwirkung verschwindet dadurch auch das lästige Sirren der eingebauten Spannungswandlerspule. Der Vorspannungsregler läßt sich dabei ganz einfach aufbauen, wie nachstehender Schaltplan zeigt:



Beim Aufbau der Schaltung sind die Anschlußdrähte zu Ösen zu biegen. Die Abblockkondensatoren können an einer Lötöse befestigt werden, die an der Kühlfahne des ICs befestigt wird. Damit die ganze Mimik nicht zu heiß wird, sollte das IC auf ineinandergestellte Fingerkühlkörper (57x57 und 42x42 mm) geschraubt werden.

G. Pfeifer, Regensburg

Layout aus dem Mikro: Fehlgeroutet

An Eurem an sich sehr interessanten Artikel ist mir eine Unsauberkeit unangenehm aufgefallen:

S. 60 'Rückwärts routen': Gehe ich wie das Programm nach dem Schema: oben, unten, links, rechts vor, so finde ich den folgenden Weg (Bild 3e, S. 59)

```
                2
            2   1   2
        2   1   3   1   2
    2   1   3   2   3   1   ⊕
    1   3   2   1   2   3   1
1   3   2   1   ⊕—1—2—3   1
    1   3   2   1   2   3   1
        1   3   2   3   1
            1   3   1
                1
```

und nicht den von Euch im Bild 3e veröffentlichten. Der veröffentlichte Weg gefällt mir aber besser, was nun?!

Bernhard Beyhl, Frankfurt

Formel falsch

In diesem Artikel hat 'Murphy' ganz fürchterlich zugeschlagen!

Man berechne die Speicheradresse für x = 255 und y = 127 und MTX0 = 2000_H! Sie lautet: $11F7F_H$! (5 Stellen!!!) Grund: **die Formel ist falsch!**

Sie lautet richtig:
XYADR =
MTX0 + **X** + (**Y** * X_{max})

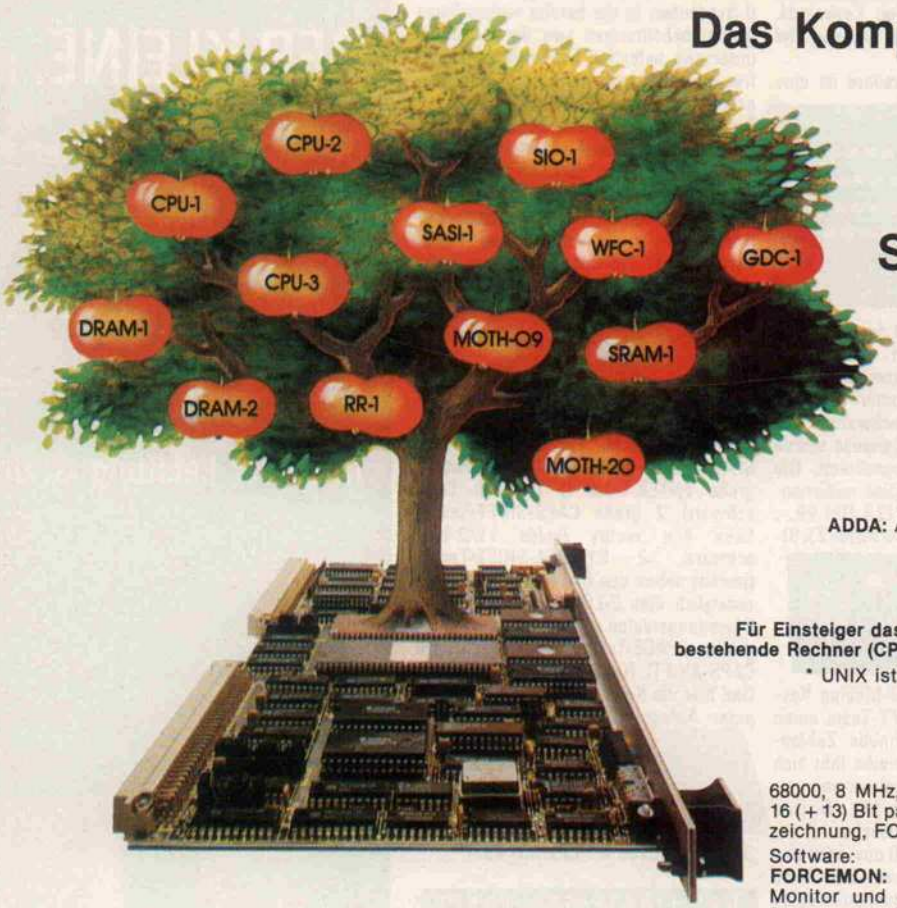
Auch im Programm müssen die Register entsprechend getauscht werden.

Bitte etwas mehr Sorgfalt bei der so unangenehmen Theorie; sonst geht's meistens schief!

Uwe Kloß, Braunschweig

Beide Leser haben recht. (Bis auf die Tatsache, daß Murphy mit diesem Versehen nichts zu tun hat.) (Red.)

E.V.G. liefert



Das Komplettsystem — VME Bus

CPU: 68000 8(10) MHz, CPU-1 Karte, 640 KByte RAM / 256 KByte EPROM, 20 MByte (form.) Platte & 1 MB Floppy

Betriebssystem: COHERENT* (Superset von UNIX*) Multiuser / Device indep. I/O

pSOS-68h: Echtzeitbetriebssystem

Sprachen: ASSEMBLER, BASIC, C, FORTRAN77 (z. Zt.)

SYS68K VME Bus Karten:

CPU-1: 68000, 8(10) MHz, 128 (512) KByte RAM, 32 (256) K EPROM, Uhr (1/1000 s ... Monate) mit Akku, 24 Bit Timer mit int./ext. Clock, 3 RS232 (110...38400 Bd), 16 Bit parallel I/O + 4(8) Handsh.

CPU-3: CPU Karte mit MMU für Multiprozessor

WFC-1: Winchestercontroller für 3 Drives und 4 Floppys

RR-1: 128...512 KByte CMOS RAM mit Akku und/oder ROM/EPROM/SRAM

GDC-1: Grafik, 16 Farben gleichzeitig, 1K x 1K Punkte, 512 x 512 Fenster, Grafikprozessor

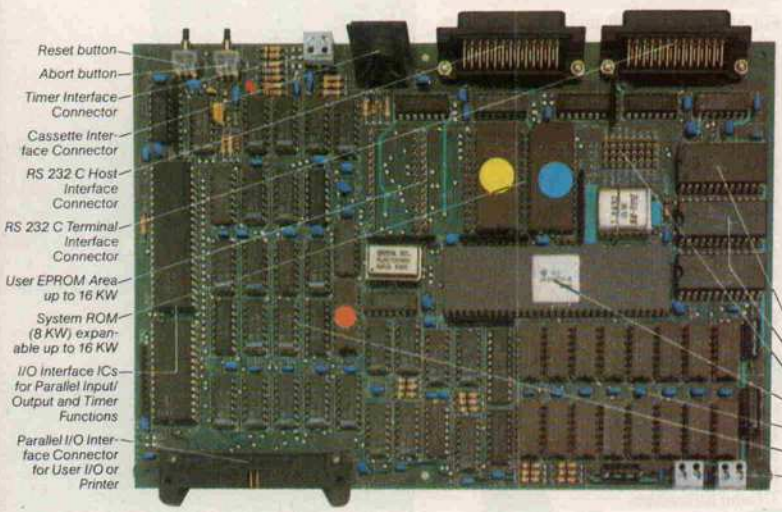
ADDA: AD-Wandler mit 32 (16 diff.) Kanälen, 12 Bit, 60 us, Eingänge per Software einstellbar, Empf. max. ±10 mV, DA-Wandler, 2 Kanäle, 12 Bit, 5 us

Weitere Karten auf Anfrage.

Für Einsteiger das preiswerteste 68000 Entwicklungssystem zum Anschluß an bestehende Rechner (CP/M Kommunikations- und Datentransfer Software verfügbar):

* UNIX ist Warenz. der BELL LABS, COHERENT von MARK WILLIAMS

FORCE Profi Kit 2



68000, 8 MHz, 128 (512) KByte RAM, max. 128 KByte EPROM, 2 x RS232, 16 (+13) Bit parallel I/O, 3 Timer, Kassettenrecorder Interface zur Datenaufzeichnung, FORCEMON im Lieferumfang

Software:

FORCEMON:
Monitor und umfangreicher Debugger mit Tracemode, Assembler- und Disassemblerfunktion

FORCEMON/IDEAL:
komfortabler Bildschirm Editor und Assembler, die zusammenarbeiten: interaktive Fehlerkorrektur, Labelnamen: 20 Stellen signifikant, Unterstützung blockstrukturierter Programmierung

BASIC-68K:
Editor (mit Such-&Ersetzfunktion), Compiler für erweitertes ANSI BASIC, 2 bis 10mal so schnell wie andere 16 Bit BASICs: z. B. leere FOR Schleife 10000mal in 1,31 sec, mit Hilfsmitteln für strukturiertes Programmieren: IF/THEN/ELSE/ENDIF, DO/UNTIL, WHILE/DO/WHEND, Typendeklaration und strenge Typenüberprüfung: REAL, DOUBLE PRECISION, INTEGER, STRING, BOOL
Zahlensysteme: Hex/Dez/Okt/Bin, Namen beliebig lang
Fehlersuchhilfen: ON ERROR GOTO, TRACE Mode, Assemblerprogramme können leicht eingebunden werden.

FORTH:
fig FORTH 79 Standard mit zusätzlichen Befehlen zur Unterstützung des großen Adreßraumes

Die oben genannte Software, deren Dokumentation in Deutsch oder Englisch geliefert werden kann, ist auch auf der CPU-1 Karte lauffähig, die man dann später zu einem VME System ausbauen kann.

Preise (incl. MwSt.)

PROFIKIT 2	1995,—	**FORTH-68K**	387,60	**SYS68K/STAND1**	
FORCEMON/IDEAL	324,90	**SYS68K/CPU1**	3876,—	**System mit 20 MByte**	19528,—
BASIC-68K	387,60	**Handbücher einzeln**	34,20		

Technischer Support für OEMs verfügbar.

Falls Sie Ihre CPU-1 oder den Profi Kit 2 selbst erweitern wollen: WD-1002 Winchester + Floppy Controller, fertiges Modul zum Anschluß von Standardlaufwerken, schreibt/liest und formatiert automatisch für nicht mehr als ..... **DM 1024,—** incl. MwSt.

Dazu passend: Slim-Line 5¼" Winchester-Laufwerk, 10 MByte formatiert ..... **DM 2995,—** incl. MwSt.

Computer für die Industrie
Computer für Institute
Computer für Amateure
Bauteile + Systemteile

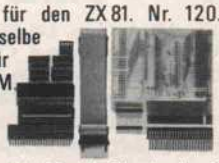
E.V.G. ELECTRONIC VERTRIEBS-GMBH
Meisenweg 10 · D-3012 Hannover-Langenhagen
Tx.: 923203 evghl d · Tel.: 0511/782078

DAS COMPUTE

HARDWARE

...und hier beginnt die Hardware-Realität.

Pio-Interface für den ZX 81. Nr. 120. DM 95,–. Dasselbe gibt es auch für den SPECTRUM. Nr. 121. DM 115,–.



Das HRG-Graphic-Modul für hochauflösende Graphik. Mit der Superauflösung von über 47000 Punkten. 16K erforderlich. Auch bewegte Displays möglich. Modul ansteckbar. Nr. 126. DM 179,–.

EazyLoad – eine tolle Erfindung. Nun gehören SAVE oder LOAD-Probleme der Vergangenheit an. Wird einfach zwischen Cassettenrecorder und Computer geschaltet. 2 Funktionen durch Schalter – LOAD oder SAVE. Nr. 127. DM 29,–.

ZX 81 ohne Grenzen mit der 23-poligen Steckerliste. Nr. 129. DM 14,50. Das gleiche Produkt für den SPECTRUM unter Nr. 119. DM 17,50.

Das 16K RAM PACK, das jeder für seinen SINCLAIR ZX 81 braucht bestellen Sie unter Nr. 125. DM 98,–.

Mit diesem Baustein erhöhen Sie die Speicherkapazität auf 32K-Byte. An der Rückseite können weitere Zusatzgeräte, wie z.B. der ZX-Printer angeschlossen werden. Dieses Gerät ist mit allen bei uns erhältlichen Modulen kombinierbar. Nr. 132. DM 149,–.

RS 232 Interface zum Anstecken an Ihren ZX 81. Kabel mit montiertem Normstecker wird mitgeliefert. Ansteuerbar in Basic- oder Maschinensprache. Kompatibel. Nr. 130. DM 198,–.

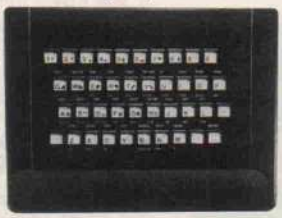
Ab sofort können Sie auf Ihrem Bildschirm und Printer mit dem ZX 81 groß und klein schreiben. Das Kabel und Modul werden gesteckt, sodaß dieses bei Nichtgebrauch leicht abzunehmen ist. Kompatibel. Nr. 131. DM 69,–.

Schluß mit allen Kassetten-Problemen macht der japanische Nobel-Kassettenrecorder. Slim & Mini mit allen computernotwendigen Funktionen wie Zählwerk, Klinkensteckerbuchse für MIC + EAR, Netzanschluß sowie Batteriefach, Batteriekontrolle durch LED-Anzeige, eingebautes Mikrofon, Pause-Taste, eingebauter Lautsprecher und natürlich alle anderen Funktionen wie Vor- und Rücklauf usw.. Nr. 122. DM 119,–.

KEYBOARDS

Die Problemlöser unter den Keyboards. Stundenlanges und sicheres Arbeiten und viel Spaß.

Das Standard-Moving-Keyboard ist eine Neuentwicklung auf dem Tastaturenmarkt. Die Belegung der Tasten entspricht exakt der SINCLAIR ZX 81-Folientastatur. Kein Löten, kein Basteln, einfach nur einstecken – und schon ist Ihr Microcomputer betriebsbereit. Ein formschönes, schwarzes und ergonomisch gestaltetes Gehäuse wurde auf das ZX 81-Design abgestimmt. Die Tasten sind auf ihre Funktion millionenfach geprüft. Nr. 123. DM 98,–.



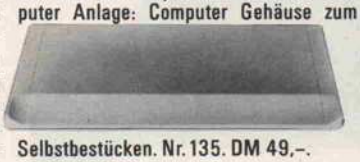
Außer der normalen ZX 81-Tastatur besitzt das Super-Moving Keyboard eine zusätzliche SHIFT-Taste, einen 10er Block für eine schnelle Zahleneingabe. Die obere Tastenreihe läßt sich auf Dauerfunktion umschalten; mit nur einem Tastendruck können Sie z.B. eine komplette Programmzeile löschen. 2 zusätzliche Tasten, um die PIO aus- oder einzuschalten. (High oder Low). Eine Taste können Sie nach Ihren Wünschen belegen. Leichte Lötarbeit erforderlich. Nr. 124. DM 175,–.

Diese Tastatur kann direkt nach Abnehmen der Originalblende und der darunterliegenden Silikontastenmatte ausgetauscht werden. Zum Lieferumfang gehört die komplette Aufsatztastatur und die Original-SINCLAIR-Beschriftung, die auf die Tasten geklebt und mit transparenten Abdeckungen versehen wird. In dieser Reihenfolge werden die Bauteile ganz einfach auf die Leiterfolie aufgebaut und mit 6 Schrauben in die bereits vorhandenen Aufnahmebohrungen von der Gehäuseunterseite befestigt. Große, bedienungsfreundliche Tasten erleichtern das Programmieren. Nr. 133. DM 98,–.

Diese formschöne und benutzerfreundliche Tastatur besitzt außer allen SPECTRUM Funktionen darüberhinaus noch viele weitere Vorzüge. Der Anschluß ist denkbar einfach, da die 2 Flachbandkabel-Anschlüsse in die vorhandenen Steckkontakte der Folientastatur eingesteckt werden. Kein Löten erforderlich! Hier

einige technische Einzelheiten: Große SPACE-Taste (8-fach Taste schwarz), große ENTER-Taste (1 1/2-fach Taste schwarz). 2 große CAPS-SHIFT-Tasten links und rechts (beide 11/2-fach schwarz), 2 SYMBOL-SHIFT-Tasten (jeweils neben den CAPS-SHIFT-Tasten), zusätzlich eine E-LOOK-Taste, CURSOR-Bewegungstasten zusätzlich neben der großen SPACE-Taste (in Verbindung mit CAPS-SHIFT). Nr. 134. DM 198,–.

Und hier die Kompakt-Idee für Ihre Computer Anlage: Computer Gehäuse zum Selbstbestücken. Nr. 135. DM 49,–.



DAS EINSTEIGER-PAKET

Für alle, die die Welt und die Faszination der Microcomputer erleben wollen und natürlich für alle Computer-Fachleute haben wir ein SUPER-EINSTEIGER-PAKET geschnürt. Unter der Bestell-Nr. 007 erhalten Sie für nur DM 498,– den kompletten ZX 81-Bausatz wie unten beschrieben und den SEIKOSHA GP 50S (Beschreibung nebenstehend). Inbegriffen ist ein 212-seitiger Basic-Kurs zugleich Ihr ZX 81-Handbuch. Auf die Komplett-Mini-Microcomputer-Anlage sind wir stolz, hier nochmals die wichtigsten Daten zum Bestellen: Nr. 007, DM 498,–.

DER SINCLAIR-ZX 81-BAUSATZ

Wir haben den ZX 81 BAUSATZ im Angebot. Für alle Einsteiger, Elektronik-Freunde und Do-it-yourself-Freaks, die Freude am Basteln und Löten haben. Denn die Montageanleitung stammt aus dem Hause SINCLAIR. Für ganze DM 129,– erhalten Sie den kompletten Bausatz mit dem Original 212-seitigen Handbuch, Netzteil, Anschlußkabel für TV und Kassettenrecorder. Und 8K-Byte BASIC ROM, 1K-Byte RAM und Z80A-CPU. Außerdem haben Sie auf alle Teile die Original SINCLAIR Garantie. Achten Sie auf unser SUPER-EINSTEIGER-PAKET – Drucker und Bausatz zu einem Super-Preis. Den Bausatz alleine bestellen Sie bitte Nr. 001, DM 129,–.

DER KLEINE. De

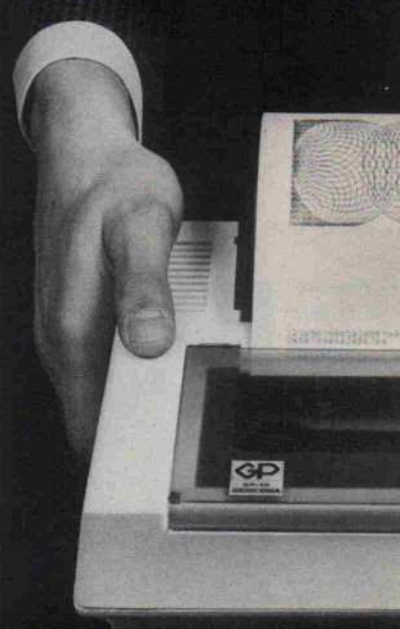
GP-50S. Genannt der ›K

kompakt. Überspielt besc

Normalpapierdrucker. Ein

Zubehör und ohne Umstä

In seiner Leistung ist der



Der Normalpapier-Drucker mit eingebautem Interface für den SINCLAIR ZX 81 und ZX-SPECTRUM 16 und 48K. Mit Sinclair Normstecker und Netzteil. Sofort betriebsbereit.

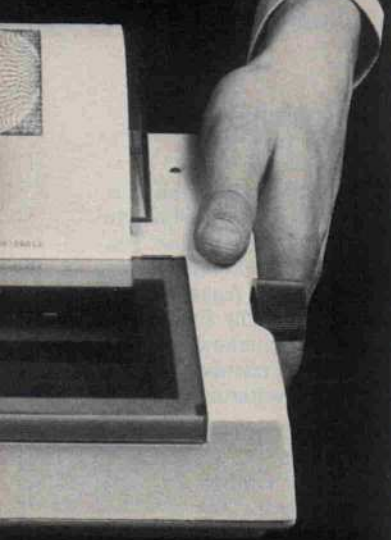
Der Friktionsantrieb gestattet die Verwendung von Rollenpapier und Einzelblatt-Papier bis zu 127 mm Breite.

Modus für Grafik, einfache und doppelte Zeichenbreite innerhalb einer Zeile möglich.

Voll grafikfähig, Normalschrift und doppelte Schriftbreite, Druckposition

R PROGRAMM

Seikosha Graphic Printer
ine‹. Handlich, praktisch,
eiden, was in ihm steckt.
ebautes Interface. Ohne
de sofort funktionsfähig.
Kleine groß. **RIESIG.**
DM 398,–



durch Zeichen oder Punkt adressierbar (Positionssteuerung).
Das Druckformat: 5 x 8 Punkt-Matrix-Druckkopf
Druckgeschwindigkeit: 40 Zeichen/s
Max. Spaltenzahl: 46 Spalten (= 322 Punkte)
Druckarten: Standardzeichen, doppelte Zeichenbreite und Grafik
Nutzen: 1 Original und 1 Kopie
SEIKOSHA GP-50S, 1 Papierrolle, Farbband, Netzteil und Handbuch
Best.-Nr. 136. DM 398,–.

COMPUTER BÜCHER

Unentbehrliche Nachschlagewerke, faszinierende Programme und jede Menge Tips und Tricks für Anfänger und Fortgeschrittene.

Hier die ZX 81-Bibliothek:
49 Explosive Spiele. Nr. 200. DM 29,80.
34 1K Super-Spiele. Nr. 201. DM 19,80.
Entdecken Sie die unendlichen Dimensionen Ihres ZX 81. Nr. 202. DM 29,80.
Das ZX 81 Buch. Nr. 203. DM 29,80.
Das ZX 81 ROM-Buch. Nr. 204. DM 39,80.

Und die SPECTRUM-Edition:
Spectrum Spektakulär. Fifty-fifty Spaß und Nutzen. Nr. 205. DM 29,80.
Spaß & Proft SPECTRUM. 60 Spiele und nützliche Anwendungen. Nr. 206. DM 24,80.
Das Spectrum Buch. Programmieren in Maschinensprache und Spielprogramme. Nr. 207. DM 29,80.
Das SPECTRUM ROM. Nr. 209. DM 39,80.
SPECTRUM ohne Grenzen. Über 100 Programme und Routinen. Nr. 208. DM 29,80.

ZX 81 – SOFTWARE

Jeder Computer ist so tüchtig wie die Software, die für ihn angeboten wird.
Hier ein Elite-Angebot für den ZX 81:
Die 4 folgenden Programme sind Profi-Anwender-Programme:
Basic-Compiler/M-Coder. Nr. 300. DM 29,50.

VU-Calc. Kalkulationsprogramm. Nr. 322. DM 59,–.
VU-File. Dateiprogramm. Nr. 321. DM 59,–.
Machine Code Test Tool. Nr. 307. DM 29,50.

Das ist unser Unterhaltungsprogramm:
Maze Death. Todesrennen. Nr. 301. DM 19,50.
Ghost Hunt. Gespensterjagd. Nr. 302. DM 19,50.
Crazy Kong. Gefährliches Abenteuer im Dschungel. Nr. 303. DM 19,50.
Tai. Invasion auf dem Staubplaneten mit viel Action. Nr. 304. DM 19,50.
Hopper. Frosch wie Frogs. Nr. 306. DM 19,50.
Cosmic Guerilla. Kosmische Banditen im Weltraum. Nr. 308. DM 19,50.
Damper/ Glooper. Engergie-Chaos/ Räuber & Gendarm. 2 Spiele. Nr. 309. DM 19,50.
Ocean Trader. Teuflische Piraten, nicht naß werden. Nr. 310. DM 19,50.

3 D black Star. Galaxis-Spiel. Nr. 311. DM 19,50.
Pioneer Trail. Wildwest Abenteuer ohne Fuzzy und Joe. Nr. 312. DM 19,50.
Asteroids. Kennt jeder. Nr. 313. DM 19,50.
Scramble. Galaxisjagd. Nr. 314. DM 19,50.
Munchees. Geister & Power Pillen steigern die Spielerpotenz. Nr. 315. DM 19,50.
Croaka-Crawla. Grüne Frösche gegen chromblitzende Trucks. Nr. 316. DM 19,50.
Defender. Kennt jeder. Nr. 317. DM 19,50.
Invaders. Die Außerirdischen mit ihren fliegenden Untertassen. Nr. 318. DM 19,50.
Galaxians & Gloops. Galaxis & Labyrinth-Spiel. 2 Spiele. Nr. 319. DM 19,50.
Schach. 6 Schwierigkeitsstufen bis zum Großmeister. Nr. 320. DM 39,50.
Flug-Simulation. Nr. 323. DM 39,50.
Weltrauminvasion. Nr. 324. DM 39,50.

SPECTRUM GALA-KOLLEKTION

Und hier die Gala-Kollektion für den SPECTRUM. Mit ausführlicher deutscher Beschreibung vom Joe:
Mit diesen Programmen können Sie professional arbeiten:
Tasword II. Das einzige wirkliche brauchbare Textverarbeitungsprogramm. Über 25 Funktionen von kursiv bis Super-Lettern. und und... Nr. 452. 69,–.
VU-3D. Entwurf. Gestaltung und Bewegung von 3-dimensionalen Körpern. Nr. 412. DM 59,50.
Adress-Manager. Adressen, Dateien, Register. Nr. 420. DM 49,–.
Machine Code Test Tool. Nr. 421. DM 49,–.
Basic-Compiler/M-Coder. Nr. 422. DM 39,50.
Editor Assembler. Nr. 425. DM 59,–.
Finance Manager. Super-Finanz-Programm mit vielen Funktionen. Nr. 428. DM 49,–.
Collector's Pack. Archivierungsprogramm. Nr. 413. DM 39,50.
Melbourne Draw. Das Super-Grafik-Programm, 16-fache Vergrößerung, individuelle Farbgebung pro Punkt. Nr. 446. DM 49,–.

Und jetzt wird gespielt mit Super-Action, toller Grafik und Sound:
Mined out. Der gefährliche Weg über die Minenfelder. Nr. 400. DM 39,50.
Space Raiders. Banditen im Weltraum. Nr. 401. DM 29,50.
Meteor Storm. Im Kampf gegen die Astro-Wolken. Nr. 402. DM 19,50.
Space Intruders. Eindringlinge aus dem Weltall. Nr. 403. DM 19,50.
4D Time Gate. In rasendem Tempo durch die Zeit-Zonen. Nr. 404. DM 39,50.
Ghost Hunt. Geisterjagd. Nr. 405. DM 29,50.
Maze Death Race. Bei diesem Autorennen lauern tödliche Gefahren. Nr. 406. DM 29,50.
Horace goes skiing. Viel Spaß mit Horace im Schnee. Nr. 407. DM 39,50.
Hungry Horace. Katz- und Mausspiel. Nr. 408. DM 39,50.

The Chess Player. Sehr stark mit 6 Schwierigkeitsstufen. Nr. 409. DM 39,50.
Planetoids/Missile. Science Fiction. 2 Spiele auf einmal. Nr. 410. DM 29,50.
Reversi. Nr. 411. DM 39,50.
Flight Simulation. Nr. 414. DM 39,50.
Psion Chess. Nr. 415. DM 39,50.
Chess the Turk. Fast unschlagbar mit Super-Grafik. Nr. 419. DM 49,–.
Astro Blaster. Kampf um die Galaxis. Nr. 426. DM 29,50.
Horace and the Spiders. Horace in den Spinnenbergen. Nr. 427. DM 39,50.
Arcadia. All-Abenteuer. Nr. 429. DM 24,50.
Zzoom. Als Abfängjäger in der unendlichen Galaxis. Nr. 430. DM 24,50.
Schizoids. Odyssee durch die Weiten der Galaxis. Nr. 432. DM 24,50.
Zip-Zap. Kolonisieren Sie die Planeten. Nr. 431. DM 24,50.
Jumping Jack. Ein lustiges und harmloses Spiel. Nr. 433. DM 24,50.
Molar Maul. Sie als todesmutiger Bakteriologe. Nr. 434. DM 24,50.
Ah Diddums. Der abenteuerlustige Teddybär im Kinderzimmer. Nr. 435. DM 24,50.
Pool. Billard. Nr. 436. DM 39,50.
Aquarius. Tauchergruppe im Einsatz gegen Mordmaschinen. Nr. 437. DM 29,50.
Magic Miner. Verschiedene Abenteuer im Wilden Westen. Nr. 438. DM 29,50.
Styx. Die abenteuerliche Reise ins Todesreich. Nr. 439. DM 29,50.
Electro Storm. Weltall-Schock. Nr. 441. DM 24,50.
Panic. Hält was der Name verspricht. Nr. 442. DM 24,50.
Light. Cycle. Ein gefährliches Spiel mit dem Licht. Nr. 443. DM 24,50.
The Hobbit. Tolkien-Spiel mit phantastischer Grafik. Nr. 444. DM 78,–.
Pentrator. Wehren Sie sich gegen die Eindringlinge. Nr. 445. DM 39,50.
ZX-USER-TAPE. Die Zeitung auf Kassette, mit ganzen Programmen und brandheißen Informationen. Nr. 453. DM 19,80.

Außerdem führen wir Programme für alle Heim-Computer wie BBC, Commodore, Dragon, Oric und viele andere. Fordern Sie unsere Software-Liste an.

COMPUTER ACCESSOIRES INT'L
Der Computer-Ausstatter.

Hier wird bestellt:

☐ per Vorausscheck
☐ per Nachnahme (zuzügl. Nachnahmegeb.)

Stück	Artikel-Nr.	Preis in DM
	Seikosha-Drucker GP 50S Nr.136	398,–
	Einsteiger-Paket Nr. 007	498,–

CT 4

Name _____
Straße _____
PLZ/Ort _____
Datum _____
Unterschrift _____
Bei Bestellungen unter DM 250,– zuzügl. Versandspesen.

COMPUTER ACCESSOIRES INT'L GMBH
Jägerweg 10 – 8012 Ottobrunn

'Wirtschaftsgipfel' in Hard- und Software

Über 6000 Aussteller auf der Hannover-Messe erwartet

Wer technologische und handelspolitische Trends aufspüren, Chancen und Risiken neuer Märkte analysieren und vor allem Fachwissen aktualisieren will, für den ist die Hannover-Messe ein Pflichttermin. Als 'Wirtschaftsgipfel' in Hard- und Software bietet die Hannover-Messe eine hohe Informations- und Orientierungsqualität, auch 1984 (in der Zeit vom 4. bis 11. April) wieder im Hannover-typischen Profil von 'zehn Fachmessen zur selben Zeit am selben Ort'.

Die bislang vorliegenden Beteiligungsdaten unterstreichen den erneut hohen Erwartungshorizont: gerechnet wird mit insgesamt über 6000 ausstellenden Unternehmen und Organisationen aus nahezu 50 Staaten. In den 23 Hallen und im Freigelände des größten Technologieparks der Welt wird sich die Netto-Ausstellungsfläche gegenüber dem Vorjahr um fast 10000 qm auf 412000 qm erhöhen. Dabei reicht das Technologiespektrum von neuen Energietechniken, dem thematischen Schwerpunkt der Hannover-Messe '84, über Sensorbildschirme oder die 'sprechende Zukunft' mit neuen Sprachausgabe-Systemen bis zu den jüngst gewonnenen Erkenntnissen für die Fabrikation im Weltraum.

Das CeBIT-'Centrum der Büro- und Informationstechnik' — im Zentrum des Interesses der Mikrocomputerbranche — wird erneut erweitert und belegt in diesem Jahr rund 127000 qm Ausstellungsfläche.

Noch attraktiver als im Vorjahr präsentiert sich die 'microtronic — Innovationszentrum Mikroelektronik' in Halle 12. Rund 300 Unternehmen aus dem In- und Ausland unterstreichen die Schubkraft der Mikroelektronik als einer Schlüsseltechnologie, an deren Einsatz die Wettbewerbsfähigkeit ganzer Nationen gemessen wird. In einer Sonderschau 'Innovative Anwendungen der Mikroelektronik' geht es darum, mit Produktbeispielen die vielfältigen Möglichkeiten der praktischen Umsetzung aufzuzeigen.

c't finden Sie übrigens gleichfalls in der microtronic-Halle 12, Stand-Nr. 3200.

Tageskarte im Vorverkauf 17 DM

Der Karten-Vorverkauf für die Hannover-Messe läuft auf vollen Touren. Im Vorverkauf kostet die Eintrittskarte für die zehn Fachmessen für einen Tag 17 Mark (Tageskasse: DM 22,—), die Dauerkarte 45 Mark (Tageskasse: DM 55,—).

Z800 — wann er kommt

Die schon vor zwei Jahren angekündigte, bis heute aber nicht lieferbare CPU Z800 soll nun im III/IV. Quartal 1984 in Mustern zur Verfügung stehen. Produktionsstückzahlen sollen ab dem ersten Quartal 1985 lieferbar sein.

Informationen:
ZILOG GmbH
Eschenstraße 8
8028 Taufkirchen

3600 Druckzeilen pro Minute

Ein neuer IBM-Zeilendrucker kann entweder mit 132 oder 168 Druckpositionen je Zeile arbeiten. Mit einer wählbaren Zeilenleistung von 2200, 3000 oder 3600 Zeilen pro Minute ist er der schnellste mechanische Drucker aus diesem Hause. Der Preis für den 'schnellen mechanischen' beträgt 210000 D-Mark zuzüglich MwSt.

Informationen:
IBM Deutschland GmbH
Postfach 80 08 80
7000 Stuttgart 80

Mehr Speicher für IBM PC/XT

Die Speicherkarte 'ISAAK' für den IBM PC ist von 64 KByte bis 512 KByte ausbaufähig. Ein besonderes Merkmal dieser Platine ist nach Angaben des Herstellers, daß ihr Preis trotz doppelter Speicherausbaukapazität dem Preisniveau der 256 KByte-Erweiterungskarten entspricht.

Informationen:
ISI Computer GmbH
Otto-Hahn-Straße 34
8012 Ottobrunn

Floppy für Apple II

Speziell für den Apple II und Apple IIe bietet die Firma nbn Floppy-Stationen an, die sich durch die halbe Bauhöhe der Apple Disk II Laufwerke und eine zwei- bis viermal so schnelle 'Track to Track'-Zeit auszeichnen.

Informationen:
nbn Elektronik GmbH
Gewerbegebiet
8036 Herrsching

64KBit E²PROM in CMOS

National Semiconductor hat jetzt die Spezifikationen des 64-KBit-microCMOS-E²PROMs bekanntgegeben. Der elektrisch löschbare Speicherchip arbeitet mit einer Versorgungsspannung von 5 V und bietet bei geringer Leistungsaufnahme eine hohe Arbeitsgeschwindigkeit. Mit der Einrichtung des 'Daten-Polling' kann das E²PROM einem Signalprozessor mitteilen, wann eine Schreiboperation ausgeführt ist.

Informationen:
National Semiconductor
Industriestraße 10
8080 Fürstenfeldbruck

Ferien mit dem Computer

Für alle jugendlichen Computer-Fans ab dem 7. Schuljahr führt die GISE einen speziellen EDV-Kurs während der Oster- und Sommerferien durch. Ziel des ein- bis zweiwöchigen Kurses ist es, Kenntnisse in der Programmiersprache BASIC zu vermitteln. Ebenfalls in der Ferienzeit werden Kurse für Erwachsene angeboten, deren Ziel die Vermittlung von EDV-Kenntnissen für die praktische Umsetzung im Arbeitsalltag ist.

Informationen:
GISE
Postfach 12 40
8740 Neustadt/Saale

Intelligente Winchester-Drives

Die Winchester-Laufwerke der Serie 700S von Shugart vereinigen die Intelligenz einer SCSI-Schnittstelle mit der Leistungsfähigkeit von Winchesterlaufwerken. Dadurch ergeben sich, neben der Vereinfachung der Systemintegration, auch Kostenreduzierungen, da Laufwerk und Controller nicht mehr getrennt ausgetestet werden müssen. Die Laufwerke werden mit einer Kapazität von 5 und 10 MByte geliefert.

Informationen:
Shugart Associates GmbH
Drygalski-Allee 33
8000 München

EIN NEWCOMER MIT 13 JAHREN ERFAHRUNG

Aus der Hazeltine Terminal Division entstand Esprit Systems Inc. So kann dieses Unternehmen, obwohl ein neuer Name auf dem Elektronikmarkt, doch auf die Erfahrung eines bekannten Terminal-Herstellers verweisen. Aufbauend auf einer bewährten Produktpalette wird Esprit Neuentwicklungen und einen optimierten Kundendienst anbieten. Sie haben also die Auswahl unter einem breiten Spektrum leistungsfähiger Geräte: IBM-kompatible Terminals ebenso wie flexible OEM-Systeme mit farbigen oder monochronen Bildschirmen. Egal, für welche Version Sie sich letztendlich entscheiden: Sie bekommen ein Produkt, das sich durch ein bestmögliches Preis-/Leistungsverhältnis auszeichnet. So z. B. unser neues, DEC-kompatibles Terminal Executive 10/102, das eine neue Dimension ergonomischen Designs in seine Preiskategorie einbringt.

SYNELEC
DATENSYSTEME GmbH

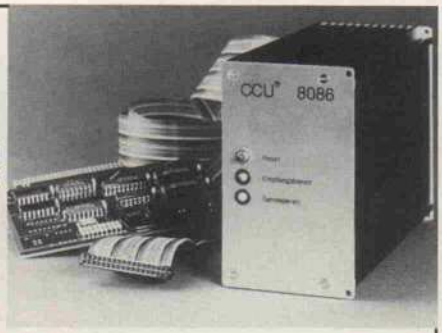
Lindwurmstr. 117 · 8000 München 2
Telefon 0 89/7 25 30 81 · Telex 5 212 289 syn d
Geschäftsstelle Düsseldorf: Telefon 02 11/35 02 36
Geschäftsstelle Frankfurt: Telefon 0 60 81/4 21 55

Autorisierter Distributor für die Bundesrepublik Deutschland

COMMODORE und APPLE
als 16-bit-Rechner mit 1024K-Speicher

CCU 8086-Central Computer Unit

Erweiterungseinheit
mit 16-bit µP 8086 — incl. bis zu 1024 KB großem Speicher — für alle 8-bit-Computersysteme.



Standardausführung bestehend aus:
CCU 8086 mit µP 8086 (5 MHz) und 128 KB Speicher (erweiterbar auf 1 MB), Interface-Karte passend für Apple II, Commodore, IBM-PC (TRS-80 und ECB-Bus in Vorbereitung), Verbindungskabel 34polig ca. 80 cm.
Gehäuse: 170 x 110 x 67 mm (geeignet für zweite CCU- oder Netzteil-Karte) und Diskette mit: Bootsoftware für CP/M 86 mit RAM-Disk 16 bit. Preis incl. Mehrwertsteuer DM 1360,—

Anwendung als:
1. 16-bit-Maschine
Hauptrechner für alle 16-bit-Programme, z. B. CP/M 86, MS-DOS 2.0 usw. Der 8-bit-Host-Rechner (Apple, Commodore, IBM-PC usw.) wird in diesem Falle als Terminal mit Floppy-Disk verwendet.
2. RAM-Disk 8-bit
definiert den CCU-Speicherbereich als Disk-Laufwerk im bisherigen 8-bit-Computer (z. Z. für Apple DOS, Apple Z80, ITT 3030 lieferbar).
3. RAM-Disk 16-bit
Der Speicherbereich über 128 KB kann auch als RAM-Disk verwendet werden.

— Software für RAM-Disk 8-bit (Apple DOS oder CP/M 80) ..... DM 98,—
— CCU-RAM-Erweiterungssatz je 128 KB ..... DM 390,—
— CCU komplett mit 512 KB Speicher ..... DM 2390,—
— CCU komplett mit 1 MB Speicher ..... DM 4480,—
— Handbuch MS-DOS 2.0 in Deutsch ..... DM 120,—
— Betriebssystem CP/M 86 auf Ihrem 8-bit-Format ..... DM 980,40
— MS. DOS 2.0 auf Ihrem 8-bit-Format ..... DM 364,80
— Umformatierung einer 16-bit-Software vom 8" Floppy CP/M Standard-Format (IBM 3740) auf 5.25" 8-bit Host-Rechner-Format (andere Formate auf Anfrage) ..... DM 80,—
— Netzteil 5 V/2,7 A ..... DM 274,—
Baugruppe speziell für ITT 3030:
256-K-RAM-Karte (huckepack) für CPU Z80 ..... DM 1090,-

ancon COMPUTERTECHNIK ®
Ingenieurgesellschaft mbH
Auguste-Viktoria-Str. 63
1000 Berlin 33

CE-DATA®

Slimeline 5,25" FD Disk
250 KB – 1 MB
Einsetzbar für alle Systeme

CE-DATA Slimeline Laufwerk 5,25" 40 Track SS/DD, 250 KB	DM	590,–
Laufwerke mit größerer Kapazität, ab Lager lieferbar, Preise auf Anfrage		
CE-DATA Doppelfloppy im Gehäuse, komplett betriebsfertig, à 250 KB	DM	1.425,–
Floppy Disk Controller für Genie I, II und Tandy TRS 80 Mod. I, single + double density mit Druckerschnittstelle, betriebsfertig im Gehäuse	DM	675,–
Double Density Controller	DM	198,–
Floppy Disk Controller für Apple (universal) incl. Anschlußkabel	DM	189,–
Apple Controller für Original-Laufwerke	DM	115,–
Verbatim Disketten	ab DM	52,–
BASF Disketten	ab DM	55,–
Siemens PT 88 Tintenstrahldrucker	DM	1.750,–
Star Drucker Gemini 10 X	DM	1.145,–

CE COMPUTER ELECTRONIC GMBH
Reichshofstr. 55 · 5840 Schwerte-Westhofen
Tel. 0 23 04 / 6 80 64-65
Komplett-Katalog gegen DM 5,– in Briefmarken
Händlerkonditionen bitte schriftlich erfragen

CE-DATA Service löst Ihre Reparaturprobleme!

Krise bei Victor

Produktion künftig in Europa?

Victor Technologies, Hersteller des 'Sirius' und des 16-Bit-Portables 'Vicki', steckt in einer tiefen Finanzkrise. Ähnlich, wie vor wenigen Monaten Osborne, hat der amerikanische Computer-Hersteller ein Verfahren nach 'Chapter 11' des US-Konkursrechts beantragt, das einen vorübergehenden Schutz vor Gläubiger-Forderungen gewährt.

Victor muß jetzt innerhalb kurzer Frist einen Finanzierungsplan vorlegen, der durch eine Kommission überprüft wird. Lehnt diese den Plan ab, bedeutet das den Konkurs. Zur Zeit werden Entwicklung und Produktion, finanziert durch die Hausbank, mit einer auf 250 Mitarbeiter geschrumpften Belegschaft weitergeführt.

Die deutsche Sirius-GmbH ist von diesen Vorgängen rechtlich nicht betroffen. 'Wir sind finanziell völlig gesund und schreiben weiterhin schwarze Zahlen', hieß es aus der Frankfurter Unternehmenszentrale. Man erwägt dort, im 'Ernstfall' die gesamte Produktion nach Europa zu verlegen.

Victor hat nach Angaben aus Frankfurt auf dem deutschen Markt bisher rund 6000 'Sirius'-Computer verkauft. Die Nachfrage nach dem Ende 1983 vorgestellten 'Vicki'-Portable sei 'ungeheuer groß'. Dieser Computer war freilich bei Redaktionsschluß noch nicht lieferbar, doch erwartete man, daß die Auslieferung noch im ersten Quartal 1984 beginnen werde.

64 KByte im EPROM

Das erste EPROM mit 512 KBit Speicherkapazität ist jetzt in Musterstückzahlen verfügbar. Der Am27512 ist in 65536 Worten zu 8 Bit organisiert und hat eine Zugriffszeit von 250 ns. Er zeichnet sich durch eine geringe Leistungsaufnahme, 132 mW in Bereitschaft, 525 mW im Betrieb, aus.

Informationen:
Advanced
Micro Devices GmbH
Rosenheimer Straße 139
8000 München 80

'Hacker-König' berät Hersteller

Der international bekannte 'Hacker-König' Richard Cheshire macht zur Zeit als Propagandist für die 'Gegenseite' von sich reden. Cheshire, der in der Bundesrepublik durch ein Spiegel-Interview bekannt wurde, tritt auf einer Fachtagung des Leuro-Seminars zum Thema 'Manipulationstechniken der Computerkriminalität und Sicherungskonzepte' auf. Er demonstriert dort live, wie man fremde Computer-Rechenzentren 'elektronisch knacken' kann. Nach Angaben der Veranstalter gibt er den versammelten Experten auch Hinweise, welche Schutzmaßnahmen von den Unternehmen getroffen werden können.

Transistor Array mit Uce = 50 V

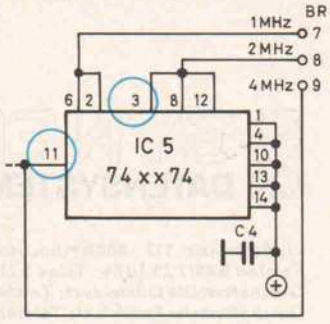
Toshiba stellt ein Transistorarray vor, in dem fünf einzeln isolierte NPN-Transistoren in einem 16poligen DIP-Gehäuse untergebracht sind. Die Transistoren sind für eine Uce von 35 V oder 50 V bei einem I_{cmax} von 200 mA ausgelegt. Alle Anschlüsse der Transistoren sind einzeln herausgeführt.

Informationen:
Bitronic GmbH
Dingolfinger Straße 6
8000 München 80

Ergänzungen + Berichtigungen

CEPAC-65 (3/84)

Im Schaltplan des c't-Projekts 'CEPAC-65' in Heft 3/84 ist uns leider ein Fehler unterlaufen. Das Platinenlayout ist jedoch korrekt. Hier also der berichtigte Teil des Schaltbildes:



»Cromemco lehrt Computer Sehen.«

Neuer Video-Digitalisierer setzt neue Maßstäbe

Die preiswerte Lösung zur aufwendigen Computergrafik heißt „Cromemco-Video-Digitalisierer".

- Farbbilder, mit normaler Videokamera aufgenommen, lassen sich „digitalisiert" im Rechner verarbeiten
- Bild- und Textkombinationen sind jederzeit möglich
- Archivierung und Speicherung von Bildern, z. B. für Inventuren, Bildkataloge etc.
- Durchführen strenger Sicherheitskontrollen
- Einblenden von Computergrafik in Original-Videobilder, z. B. im Training und bei Lehrgängen

Übrigens, alle anderen EDV-Anforderungen erfüllen Cromemco Systeme selbstverständlich auch.
Wie können wir Ihnen helfen?
Fordern Sie uns!

Cromemco®

Cromemco GmbH · Frankfurter Str. 33–35 · 6236 Eschborn 1 · Tel. 0 61 96/48 16 06

Mittwoch, 4. – Mittwoch, 11. April
Hannover Messe'84
CeBIT Halle 4, Stand 1903–2003

Informations Coupon c't 4

Ich erbitte Unterlagen über Cromemco-Produkte:
☐ Mikrocomputer-Systeme ☐ Video-Digitalisierer ☐ Personal Computer C–10 ☐ S–100 Boards

Name: _____

Straße: _____

PLZ/Ort: _____

Tel.: _____

Cromemco GmbH · Frankfurter Straße 33—35 · 6236 Eschborn 1 · Telefon 0 61 96/48 16 06

Heise Software

Superhits für VC 20 und C 64!

c't-Programme

Alle in den Rubriken 'c't-Programme', 'c't-Projekt' und 'Computer zu Hause' vorgestellten Programme sind auf Datenträgern erhältlich.

Zur Zeit sind lieferbar:

Nr.	Programm	Datenträger	Preis
S831241	MINIMON (Z 80-Monitor)	Kassette (TRS-80)	5 DM
S831241	Terminal-Betriebsprogramm	EPROM (2732)	20 DM
S831242	Terminal-Zeichensatz ZS0	EPROM (2732)	20 DM
S831243	Zeichensatz ZS1 (deutsch)	EPROM (2732)	20 DM
S831272	Textbausteinprogramm	Kassette (TRS-80)	5 DM
S831292	Master-Directory	5¼-Zoll-Floppy	15 DM
S840145	c't 86-Monitor	2 EPROMs (2732A)	69 DM
S840172	65C02-Assembler in FORTH	5¼-Zoll-Floppy (Apple)	15 DM
S840232	Energiekostenberechnung für Strom und Gas mit ZX81	Kassette	5 DM
S840349	SPRITE-Editor für C 64	Kassette	5 DM
S840324	Kfz-Kostenanalyse mit Video Genie	Kassette	5 DM

NEU

(Für VC 20 mit Speichererweiterung, min. 16 KByte)

Bitte bei der Bestellung Rechnertyp angeben!

Programmbibliothek Nr. 10
TEXTY

Ein großes Textverarbeitungsprogramm mit folgenden Features:
- Einlesen und Speichern von Texten auf Kassette oder Diskette
- Text erfassen mit Anzeige von Zeilen- und Spaltenposition
- Neue Zeile einfügen
- Druckerausgabe
- Kopieren von Zeilen
- Suchen von Textstücken; mit der Funktion Ersetzen kann der gefundene Textteil durch einen neuen, wahlweise kürzeren oder längeren Text ersetzt werden.
- Voll menügesteuert

Es sind zwei verschiedene Versionen mit spezieller Druckeranpassung erhältlich:

TEXTY GP für Commodore-Drucker VC 1515, VC 1541 und Seikosha GP 80, GP 100 VC

TEXTY MX für Epson MX 80

Kassette mit Handbuch 49,— DM

Neu:

Programmbibliothek Nr. 11
ADRESSEN

Anschriften von Freunden, Verwandten, Vereinsmitgliedern, Kunden, Lieferanten werden verwaltet und in übersichtlicher Form angezeigt. In Zusammenhang mit TEXTY Adressenausdruck für Serienbriefe realisiert.

Kassette mit Handbuch 39,— DM

Neu:

Programmbibliothek Nr. 12
KARTEIKASTEN

Dieses Programm macht alle Karteikästen überflüssig. Es erlaubt die Verwaltung beliebig großer Karteien (nur durch Speichergröße begrenzt). Möglich sind
- Anlagen einer neuen Datei
- Abspeichern auf Band oder Diskette
- Einlesen bestehender Dateien von Band oder Diskette
- Sortieren nach auszuwählenden Feldern
- Druckerausgabe mit vielen Möglichkeiten.

Für jede Kartei lassen sich beliebig viele List-Ausdrucke festlegen und ebenso wie die Datensätze auf Band oder Diskette speichern. Alle Funktionen werden über Menüs gesteuert.

Kassette 49,— DM

Programmbibliothek

Im c't-Software-Service erhalten Sie ein Sortiment besonders interessanter und leistungsfähiger Programme für verschiedene Computersysteme, das ständig erweitert wird. Allen Programmen sind ausführliche Erläuterungen, zum Teil in Handbuchform, beigefügt.

Programmbibliothek Nr. 1
(für PET 2001 (ab 8 KB), cbm 3001, TRS-80 Level II)

10 lehrreiche und unterhaltsame BASIC-Programme, u. a. Schnell-Lese-Training. Übung für das Präzisionsschreiben, Drill für das Kopfrechnen, Berechnung von Zinseszinsen, der Computer als Hellseher.

Programmkassette 19,80 DM
Handbuch (56 Seiten) allein 8,80 DM

Programmbibliothek Nr. 2
(für PET 2001 (ab 8 KB), cbm 3001, TRS-80, Level II)

10 BASIC-Programme, u. a. Drillprogramm für das Bruchrechnen, Übung für das Geschwindigkeitsschreiben, Tilgungsplan für ein Darlehen, Reaktionszeit-Test, Gedächtnis-Training, Trainingsprogramm für die Beobachtungsgabe, der Computer als Post.

Programmkassette 19,80 DM

Programmbibliothek Nr. 3
RHINO
(für PET 2001 (ab 8 KB), cbm 3001, TRS-80 Level II)

Ein spannendes Spiel für intelligente Leute. Mit vielen Variationsmöglichkeiten.

Programmkassette 19,80 DM

Programmbibliothek Nr. 4
Analog-Uhr/Digital-Uhr
(für PET 2001 (ab 4 KB) und cbm 3001)

Programmkassette 19,80 DM

Programmbibliothek Nr. 5
Morse-Tutor
(für PET 2001 (ab 8 KB), cbm 3001)

Übungsprogramme für das Erlernen des Morse-Codes. Die akustische Ausgabe erfolgt mit Hilfe eines anzuschließenden Radios oder Kassettenrekorders.

Programmkassette 19,80 DM

Programmbibliothek Nr. 6
PACK/UNPACK
(für PET 2001 (ab 8 KB) und cbm 3001)

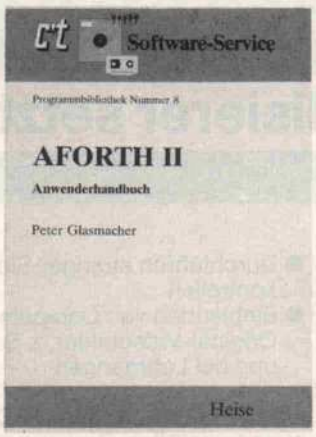
Ein sehr nützliches Dienstprogramm zum Anlegen, Ändern/Ergänzen und Lesen von Dateien aus numerischen Daten.

Programmkassette 19,80 DM

Programmbibliothek Nr. 7
Interaktive Menüplanung
(Mind. 32 KByte oder Diskette. Fordern Sie unseren Spezialprospekt an.)

Gesünder essen mit Computer-Hilfe: Das Programm enthält für die meisten gängigen Lebensmittel (fast 400) Informationen über Energie-, Nährstoff-, Mineralstoff- und Vitamingehalt. Es ermöglicht die Zusammenstellung von Mahlzeiten im Dialog mit dem Computer. Ideal für alle, die beruflich mit dem Erstellen von Speiseplänen zu tun haben, aber auch für den privaten Haushalt. Wahlweise Druckerausgabe.

Diskette oder Kassette 92,50 DM



Unser Bestseller:

Programmbibliothek Nr. 8
FORTH mit 65C02-Assembler
(für Apple und Apple-kompatible Computer mit Diskettenlaufwerk)

Das Programm enthält neben einem FORTH-Compiler nach dem FORTH-79-Standard einen zeilenorientierten Editor und einen Assembler für den erweiterten Befehlssatz der CMOS-CPU R65C02. Wenn das System mit einer 80-Zeichen-Karte ausgestattet ist, steht zusätzlich ein komfortabler Screen Editor zur Verfügung.

In 64-KByte-Systemen wird FORTH in die Language-Karte geladen und belegt den Adreßbereich (H) D000...F7FF. Die Transient Program Area (TPA) beginnt bei (H) 5000, so daß für High-Resolution-Anwendungen noch eine Seite frei bleibt. Bei anderen Systemen wird FORTH ab (H) 5000 geladen. Es steht dann mehr als 10 KByte Speicherraum für Anwenderprogramme zur Verfügung — wesentlich mehr als bei herkömmlichen FORTH-Systemen.

Der Compiler wird auf einer Diskette (Format: Apple Standard) geliefert, deren Rückseite das Source Listing des Assemblers und des Editors sowie nützliche Utilities wie einen FORTH-Decompiler und einen Textformatierer enthält. Es ist geplant, nach Festlegung des FORTH-83-Standards ein Anpassungsprogramm anzubieten.

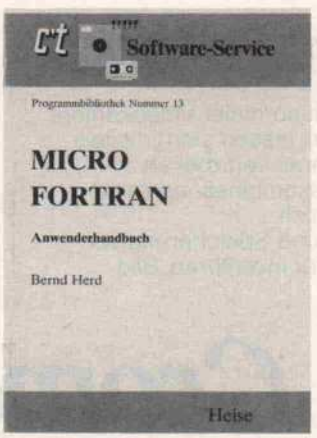
Diskette mit Handbuch 98,— DM
Zwei Disketten (single sided) mit Handbuch 113,— DM

Neu:

Programmbibliothek Nr. 9
MYSTERY
(für ZX 81)

Ein Spiel, das nie langweilig wird. Im Mystery-Land wimmelt es von Kobolden, Geistern und ähnlichen unangenehmen Zeitgenossen. Steuern Sie Ihre Spielfigur mit den Cursor-Steuertasten durch das Labyrinth. Der Computer wird Ihnen immer neue Aufgaben stellen, die Sie zu bestehen haben. Bei jedem Spiel ändern sich fast alle Variablen, so daß Sie immer neue Abenteuer erleben.

Kassette 19,80 DM



Neu:

Programmbibliothek Nr. 13
MICRO FORTRAN
(für TRS 80, Video Genie)

Micro Fortran ist ein Fortran-System für den TRS-80/Video Genie mit mindestens 16 K RAM und benötigt **keine** Diskettenstation. Da Fortran eine sehr umfangreiche Sprache ist und der Micro Fortran schon ab 16 K RAM arbeiten soll, enthält Micro Fortran nicht alle Möglichkeiten von Fortran IV. Trotzdem versteht das System die wichtigsten Fortran-Befehle, beherrscht Realzahlenverarbeitung und hat einen bequemen, bildschirmorientierten Editor. Im Vergleich zu BASIC ist Fortran wesentlich schneller, strukturierte Programmierung mit Unterprogrammen ist einfacher usw. Nachteilig ist allerdings, daß das compilierte Programm zwar sehr viel schneller ist als ein BASIC-Programmm, aber dafür auch wesentlich mehr Speicherplatz verbraucht. Außerdem muß für Fortran immer der Quelltext UND das Objektprogramm im Speicher stehen.

Das gesamte Fortran-System einschließlich Editor und Laufzeitsystem benötigt knapp unter 8 K Byte, es bleibt dem Benutzer also selbst bei nur 16 K noch genügend Platz, um einfache Programme zu schreiben. Trotzdem sind 32 K oder mehr empfehlenswert.

Das Handbuch enthält eine Einführung in den Umgang mit FORTRAN und eine ausführliche Beschreibung aller unter MICRO FORTRAN verfügbaren Befehle.

Kassette und Handbuch 70,— DM

Neu:

Programmbibliothek Nr. 14
OTHELLO
(für Apple mit Pascal)

Das Strategiespiel Othello (Reversi) in einer schnellen Pascal-Version. Drei Spielstärken sind einstellbar. Das Handbuch enthält das Listing mit sehr ausführlicher Beschreibung und ist deshalb besonders interessant für Pascal-Anfänger.

Diskette (5¼-Zoll) mit Handbuch 30,— DM

So können Sie bestellen:

Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck oder einen von Ihrer Bank quittierten Einzahlungsbeleg über die Bestellsumme zuzüglich 3 DM (für Porto und Verpackung) bei. Bei Bestellung aus dem Ausland muß stets eine Überweisung in DM erfolgen. Die Überweisung und Ihre Bestellung richten Sie bitte an:

c't-Versand
Verlag Heinz Heise GmbH
Bissendorfer Straße 8
3000 Hannover 61
Konto-Nr. 93 05-308,
Postscheckamt Hannover

Microcomputer total
im Regionalbereich HANNOVER

Das Einplatzsystem
Betriebssysteme: CP/M-80, CP/M-86, MS-DOS, UCSD (P)
Hauptspeicher: von 64 KB - 512 KB
Externe Speicher: 2 Diskettenlaufwerke (je 320 oder 640 KB), (integriert)
1 Platten-Laufwerk, Winchester, 5¼ Zoll, 10 MB (integriert)
oder bis zu 3 Platten-Laufwerke je 10 MB

Das "einfache" Mehrplatzsystem
Zwei Einplatzsysteme kommunizieren unter dem Betriebssystem MP-M miteinander

Das lokale Netz (NCR DECISION NET)
Austausch von Daten u. Programmen zwischen den Benutzern von Personal-Computern (auch verschiedenen Typs)
Gemeinsamer Zugriff auf Massenspeicher und gemeinsame Nutzung leistungsstarker Peripherie
Zugriff auf zentrale Dateien von EDV-Systemen
bis zu 64 Geräte anschließbar, wobei 16 gleichzeitig kommunizieren können

Der Supermicro NCR TOWER 1632
Prozessor: 16/32 -Bit- Motorola 68000 (Taktfrequenz 10 MHz)
Hauptspeicher: 512 KB bis 2 MB
Plattenspeicher: 28 MB bis 1000 MB (5,25-/8 Zoll Winchester-Platten)
Disketten: 5,25 Zoll bis zu 1 MB
Magnetbänder: Streaming oder externes Tape
Betriebssystem: UNIX™
Programmiersprachen: C, Cobol, Basic, Pascal, Fortran
Datenbank: INGRES
Anschlüsse: Multibus (IEEE-796)
Datenkommunikation: RS 232 C/V.24, ASCII TTY, BSC 2780/3780, NCR DLC, SDLC/SNA, X.21/X.25
Bildschirmarbeitsplätze: bis zu 16 (lokaler und externer Verbund)

Wir sind **die** Hard-und Software-Spezialisten.

Unser Leistungsangebot Software:
Standardpakete (Text, Lohn-und Gehalt, FIBU, Auftragsverwaltung) umfangreiche Branchenlösungen
Datenbanken (d'Base II, Nukleus, Ingres)
Spezialanwendungen: 3270-Emulation, SNA, usw.

Sie finden uns auch auf der Hannover-Messe, NCR-Stand, CEBIT-Halle

meyer & hanelik

Tannenbergallee 4c · 3000 Hannover 1
Telefon 0511/67 10 70

Das runde Elektronik Programm

Jetzt im neuen kostenlosen Elektronik-Spezial-katalog!

Meßtechnik ● Bauteile ● Funk ● HiFi ● Modellbau ● Bausätze ● Industrie-Restposten ● Schlagerangebote ● Das alles und mehr im 28 000-Artikel-Programm von Europas größtem Elektronik-Spezialversender.

kostenlosen Spezialkatalog anfordern!

CONRAD ELECTRONIC

Fach 698
D-8452 Hirschau

Tel.02262-4815

Endlich!

Praktische EDV-Programme für den Kleinbetrieb

Angebote, Rechnungen etc. nach Textbausteinen mit Preis, Lohn, Stück, Auf- o. Abschlag	ab DM 980,–
einf. Vor- und Nachkalkulation	DM 850,–
Finanzbuchhaltung	DM 1660,–
Lohnbuchhaltung (i.V.)	DM 1050,–
Brieftext m. Adresswesen	DM 475,–
Wärmebedarf n. neuer Norm	DM 680,–

(lauffähig auf IBM, KAYPRO, GENIE u. ä.)

MC - SERVICE
5276 BIELSTEIN

Computersysteme
Monika Glitsch, Telefon 0 71 52 / 4 76 43

Arbeitsplatzcomputer
Home-Computer
Drucker
Speichererweiterungen

Software
Literatur
Zubehör
und und und

Hochleistungs-GROSSBILDPROJEKTOREN portable...

Ludwig-Finckh-Weg 31/1 7250 LEONBERG 1

Mikrocomputerkurs im Schulfernsehen (Nord III)

Mit Z80 und 68008, Vollgrafik und Roboter

Der NDR-Klein Computer

Broschüre „NDR" anfordern!

Mit Begleitbuch von R. D. Klein

ELEKTRONIKLADEN · 4930 Detmold 18
Eggestraße 70 · Tel. 0 52 32/81 71

TRS-80 / Genie / Colour-Genie

SEDIT DM 39.–
komfortabler Bildschirmeditor, volle Cursorsteuerung + Bildschirmkontrolle einfaches Einfügen + Löschen v. Zeichen

COPBIT DM 39.–
Kopierprogramm. Kopiert praktisch alle Colour-Genie-Kassetten. Auch geschützte Formate, z. B.: Preloader/Baudrateumsch.

COLBAS DM 59.–
Umfangreiche Basicerweiterung, MERGE, Disk-Basic-Funktionen, RESTORE N, Variablensuche, Listingformatierer, Packer usw. auch kombiniert mit SEDIT für nur DM 89.–!

Ähnlich für TRS-80/Genie mit Renumber!

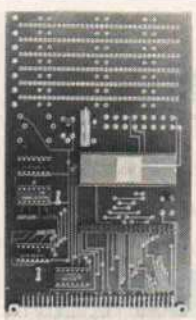
SBUG / CBUG
Monitor, Disassembler und Trace. Komfortabler Monitor mit Disassembler und progr. Trace mit Single Step durch RAM, ROM und DOS. SBUG für TRS-80/Genie DM 139.– / CBUG für Colour DM 99.–. CBUG liest auch TRS-80- und Genie-Kassetten!

Alle Programme mit ausführlicher, deutscher Anleitung, Preise inkl. 14% MwSt. SBUG/SEDIT auf Disk + 10.–. Fordern Sie Info und Hardwarepreisliste an! Händleranfragen erwünscht!

Luidger Röckrath Noppiusstr. 19, 5100 Aachen, Telefon: (02 41) 3 49 62

RATEV ELEKTRONIK-VERTRIEBS GMBH
Postfach 16 01, 4030 Ratingen 1, Tel. 0 21 02 / 2 99 02

Hochwertige Cherry-Tastatur, wie Titelbild c't Ausgabe 1, mit 8 bit parallelem Datenausgang, die an jedes System oder jeden Selbstbaucomputer angepaßt werden kann, da die ursprüngliche Version mit einem EPROM für alle Anwendungen veränderbar ist.
Eine ausführliche Umbauanleitung liegt vor.
Preis inkl. Gehäuse ..... **DM 390,––**
dto., jedoch für APPLE bereits umgerüstet ..... **DM 449,50**



MPF-1 **A/D-WANDLER-KARTE** **ECB-Bus**
16 Kanäle 8 Bit

Vielseitig verwendbar zum direkten Anschluß an das Lehrsystem Microprofessor oder Systeme mit ECB-Bus.

Best.-Nr. 4501	Bausatz MPF-1	DM 170,–
Best.-Nr. 4502	Bausatz ECB-Bus	DM 170,–
Best.-Nr. 4601	fertige Karte MPF-1	DM 220,–
Best.-Nr. 4602	fertige Karte ECB-Bus	DM 220,–
Best.-Nr. 4701	Flachbandkabel	DM 30,–

MPF-1
Kalenderuhr für Anzeige des Datums und der Uhrzeit
Ausgabeverstärker für den PIO-Baustein m. LED-Monitor

Best.-Nr. 4801	Uhr RTC-1 fertig (aufgebaut und getestet)	DM 80,–
Best.-Nr. 4801 A	RTC-1 mit Accus	DM 95,–
Best.-Nr. 4802	Ausgabeverstärker	DM 65,–
Best.-Nr. 4803	EPROM mit Uhrenprogramm	DM 20,–

Bitte Prospekt anfordern
Preise inkl. MwSt.,
Versand per NN.

CHR. BEHN Jakob-Kneip-Straße 107, 4000 Düsseldorf 13
Telefon: 02 11 / 70 70 58

PROFESSIONELLES DIRECT-DRIVE-LAUFWERK IN 19″ 28BE / 3HE CASSETTE

- Geeignet für Compact-Cassetten
- Direkt-Drive, kein Riemen
- Verschleißfreier Antrieb
- Hohe Zuverlässigkeit
- Minimale Drift- u. Flutterwerte
- Fernsteuerbar, TTL kompatibel
- Digitalaufzeichnung bis 9,6 kBd
- Analogaufzeichnung (Hi-Fi)
- Suchlauf (Cue-Review)

OPTION: 19″ Cassette (Schroff) — Leistungsteil — A-W Verstärker f. Datenaufz. — Sonderausführungen

MAGNETRONIC GmbH
Kreuzhofstr. 10, D-8000 München 71, Tel. 0 89/7 55 36 70

COMPU-SHACK

APPLE CP/M 3.00 Plus

6 MHZ z-80 läßt Ihre Programme 2–3 mal schneller laufen

Mit CBASIC, GSX-80 (Graphic Software), CP/M 3.0 + Documentation, Platine mit zusätzlich 64 k Ram; Besonderheiten: Bankselection volle 128 k, Passwortschutz; sucht Programme bis auf 4 Laufwerke; verwaltet bis zu 16 Laufwerke mit je 512 Mbyte; kein BDOS error mehr; vermerkt Datum + Uhrzeit letzten Zugriffs. **DM 1099,–**

Programmer-Kit	**DM 220,–**
Eprom Programmierkarte 24 Pin	**DM 299,–**
Z Card z-80 mit CP/M-80 Software (2.2)	**DM 375,–**
80-Zeichen-Karte	**DM 299,–**
V 24 (rs-232)	**255,–**
Centronics Printer Interface	**DM 199,–**
6 MHz z-80 b APPLI Card 64 k + CP/M 2.2 Software	**DM 899,–**
APPLI Card 64 k + 128 k + CP/M 2.2 Software	**DM 1499,–**

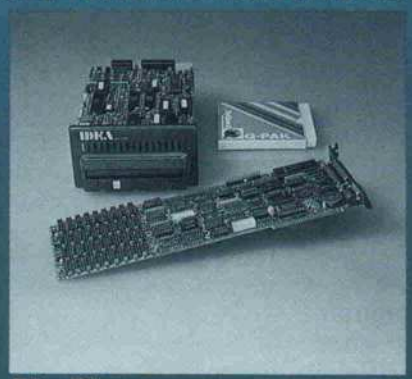
Seriell Interface **DM 240,–**
80 x Zeichen für den Apple IIe + 64 k Ram **DM 399,–**
Diskettenlaufwerk für den APPLE
Slim Line Laufwerk **DM 599,–**
100 % Apple-kompatibel, liest auch Halb-Tracks
Vista Controller für Apple 8" Laufwerke **DM 1295,–**
Teac-Laufwerk Typ FD 55A für Apple **DM 699,–**
Pinball für Apple **DM 99,–**

IBM Keyboard für den Apple
11 Funktionstasten, Nummern Tastatur **DM 775,–**
Keyboard incl. 10er Tastatur **DM 399,–**

Hercules Graphic Platine
720 h x 384 v auf Monoch.-Bildschirm **DM 1699,–**

Weitere Platinen wie zum Beispiel
68000 Chip + Unix Software für den IBM/ Ether NetLink/ 8" Controller für den IBM/ 1024 x 1024 color Graphic Carte/Speech Master/Voice Recognition/IEEE 488 Interf./ Video Platine/Mouse System

Mouse von Mouse System für den IBM **DM 1195,–**



Big Blue. Läßt sich auf Ihrem IBM 8 Bit (CP/M) Software lesen und laufen. Besitzt außerdem Calender, Clock, Battery, Harddiskadapter, Parallel, Serial Port, 64 k Ram. **DM 1995,–**
Baby Blue CP/M 2.2 mit Z 80 B und Bitam-Software **DM 1895,–**
8 x RS 232 (V24) für den IBM-PC-XT **DM 3299,–**
Accelorator mit 8086 mit 10 MHz, 128 KB aufrüstbar auf 576 KB **DM 2995,–**

Software für den IBM-PC

Lotus 1 2 3	**DM 1695,–**
Access-Manager	**DM 1095,–**
Pascal MT + 86	**DM 1095,–**

Pinball von SoubLogic für IBM **DM 149,–**
Maxell-Diskette, 10er-Pack **DM 69,–**
Microsoft Flugsimulator **DM 199,–**
IBM 3740 Format mit 5 ½" + 8"-Controller sowie einem 8"-Laufwerk mit 1,6 MB **DM 6995,–**
10-MB-Tape Backup für XT oder Winchester-Laufwerk **DM 3995,–**

Winchester Laufwerk für den IBM Intern (im Gehäuse)

12.5 MB inc. Cont. Adapt.	**DM 6995,–**
12.5 MB 2. Laufwerk	**DM 3999,–**
5 MByte Wechsellaufwerk intern für den PC	**DM 7995,–**

Festwechsellaufwerke für **IBM, Apple,** removable Harddisc von **DMA**

12 MB + 5 MB	**DM 13999,–**
20 MB + 5 MB	**DM 14999,–**
26 MB + 5 MB	**DM 16999,–**
53 MB + 5 MB	**DM 22999,–**

IBM Erweiterungsplatinen
Von A.S.T. MegaPlus **64 k bis 512 k**
S = Serial, P = Parallel, C = Clock, G = Game, H = Harddiskadapter, SP = Spooler, Software, DI = Disk-Spooler

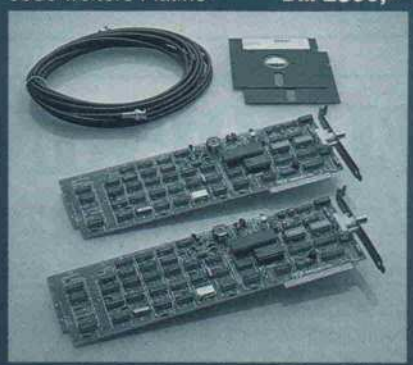
64 k, C, S, SP, DI	***DM 1190,–**
64 k, C, 2x S, SP, DI	**DM 1398,–**
64 k, C, S, P, SP, DI	**DM 1398,–**
64 k, C, S, 2x, P, SP, DI	**DM 1550,–**
Jede weiteren 64 k bis 256 k	**DM 220,–**
MegaPack 256 k bis 512 k	**DM 990,–**

Sixpack für den IBM XT, ausbaufähig bis 384 k, S, P, C, G, DI, SP **Preis auf Anfrage**

I/O Plus II

C, S, SP, DI	**DM 499,–**
jede weitere S, P oder G	**DM 199,–**
maximale Ausbaustufe C, 2 x S, 1 x G, 1 x P, SP, DI	**DM 1096,–**

Drei-Com-Ethernet Board mit 10-SB-Transfer pro Sekunde **DM 3995,–**
Drei-Com-EtherPrint Software **DM 1790,–**
PCnet Starter Kit, bestehend aus 2 Platinen, für 2 **IBM** PC oder XT, inc. Coaxial Cable, Software, Übertragungsrate 1 Mbps **DM 4990,–**
Jede weitere Platine **DM 2590,–**



Weitere Platinen von diversen Herstellern

512 k Platine mit 64 k bestückt	**DM 799,–**
jede weiteren 64 k	**DM 220,–**
Combo Carte, P, S, C, G	**DM 599,–**
Eprom Programmierkarte 4 k–64 k bit 24 Pin	**DM 455,–**

IRMA, Ihr Anschluß an den **IBM 3270** Controller **DM 3995,–**
Persyst mit **3270-3780** sowie HASP-Übertragungsprotokolle mit Koaxial-3278-Terminal **Preise auf Anfrage**
Picox 3278-Terminal mit Koaxial-Anschluß sowie up- und download-Software **DM 3995,–**

Weitere A.S.T. Platinen	**DM 5690,–**
Adv. Communication Card	**DM 799,–**
Bisync Emulation	**DM 2790,–**
SNA/SDLC Emulation	**DM 2790,–**

Tecmar Graphics Master
mit 640 h x 480 v, 16 Farben **DM 1999,–**
Graphic Master jetzt Lotus 1.2.3-fähig

Alle Preise für Industrie, Gewerbe und Handel. Endverbraucher gesonderte Preisliste einschließlich MwSt. anfordern.

Händleranfragen erwünscht
Alle Preise zzgl. der gesetzlichen MwSt.

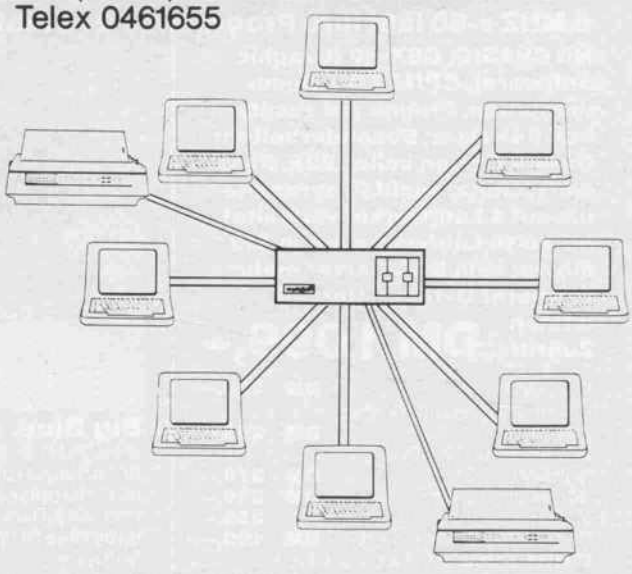
Ringstr. 56 5450 Neuwied Tel. (02631) 29031
Telex 867 716 celgk

COMPUSHACK

Syscom

NORTH STAR HORIZON
8/16 MIT TURBO-DOS

Winkler GmbH
Trübnerstraße 40
6900 Heidelberg 1
Tel. (06221) 4 91 81
Telex 0461655



Einer der zuverlässigsten Mikro-Computer für Schule, Büro, Wissenschaft, jetzt noch bestechender in Leistung und Vielseitigkeit durch:

- Multiprocessing — bis zu acht Benutzer, jeder mit eigener Zentraleinheit
- Northstar-Betriebssystem Turbo-DOS
- 8-Bit- und 16-Bit-Zentraleinheiten
- Große Speicherkapazität — bis zu 512 K-RAM pro 16-Bit-Zentraleinheit

Das HORIZON 8/16 Multi-Processor-System unterstützt bis zu 8 Benutzer.

EUROPRINT
drückt den Preis

- **MADE IN GERMANY (VDE + FTZ-Prüfzeichen)**
- Bidirektionaler Druck mit Druckwegoptimierung
- Druckgeschwindigkeit von 100 Zeichen/Sek.
- Echte Matrix mit 9x7 Punkten Druckraster daher eine hohe Auflösung von 100 Punkt/Zoll
- 9 Nadeln pro Kopf mit versetzter Anordnung zur Erhöhung der Druckdichte
- Einzelnadelsteuerung bei einer Balkenbreite von 6 Punkten vertikal hochauflösende Graphik
- 3 verschiedene Zeichenbreiten 80/100/120 Zeichen/Zeile darstellbar in doppelter Zeichenbreite (Breitschrift)
- Kursivschrift
- Unterstreichmodus
- Vollständiger Horizontal- und Vertikaltabulator beide vorwärts und rückwärts programmierbar
- Programmierbare Formularlänge
- Zeichensatz: 96 ASC II Zeichen mit Unterlängen, Standard deutscher Zeichensatz, Softwaremäßig umschaltbar auf US-Zeichensatz
- Selbsttesteinrichtung
- Ausführliche, deutsche Bedienungsanleitung
- **Zubehör:** Paralleles Interface, Centronics (**Standardausführung**)
- Serielles Interface RS 232, V 24
- Rollenhalterung
- **Formulareinzug**

UNITRONIC®

Elektronische Bauelemente und Geräte

BTX-No. 745 Bildschirmtext

UNITRONIC GMBH
Münsterstraße 338
4000 Düsseldorf 30
Postfach 330 429
Telefon 0211 / *62 63 64
Telex 858 6434
Hardwarestudio
Münsterstraße 375

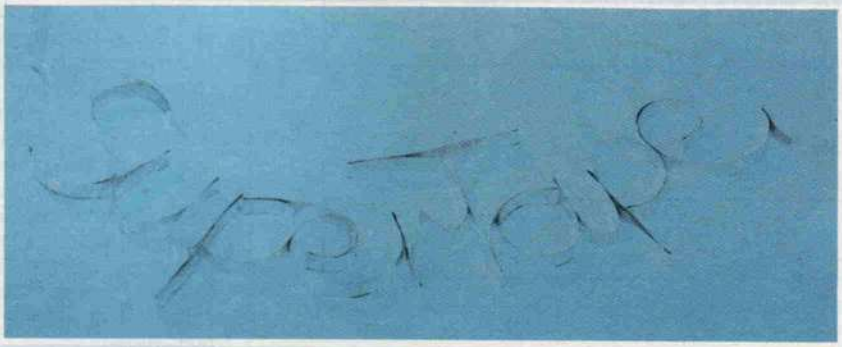
UNITRONIC HAMBURG GMBH & CO KG
Lindhofstraße 3
2360 Bad Segeberg
Telefon 04551 / 86 97 + 86 98
Telex 261 646

UNITRONIC VERTRIEBS GMBH
Manskestraße 29
3160 Lehrte
Telefon 05132 / 5 30 01
Telex 922 084

Händleranfragen erwünscht!

HANNOVER MESSE '84
Halle 1 / CeBit
Stand 7701 / 7801
ROBOTRON

Für alle Mikrocomputer:



Kassettenaufzeichnung mit 3600/7200 Baud

Andreas Stiller/Christian Persson

Dem Austausch von Daten und Programmen zwischen Computern verschiedener Fabrikate haben die Hersteller viele Hindernisse in den Weg gestellt. Eine der größten Hürden besteht in der Wahl unterschiedlicher Verfahren und Formate für die Aufzeichnung auf Datenträgern. c't hat sich die Aufgabe gestellt, dieses Problem zu lösen. Wir stellen ein schnelles und sicheres Verfahren für die Datenaufzeichnung auf Magnetband-Kassetten vor, das sich bei praktisch allen Mikrocomputern anwenden läßt. SuperTape arbeitet wahlweise mit 3600 oder 7200 Baud und ist damit schneller als alle Verfahren, die bisher von den Herstellern populärer Homecomputer verwendet werden. Zusätzliche Hardware ist nur im Ausnahmefall erforderlich.

c't stellt in diesem Monat die SuperTape-Implementation auf Z80-Rechnern am Beispiel des ZX81 vor. Lösungen für den ZX Spectrum, für TRS80, für VC-20 sowie C 64 befinden sich im Erprobungsstadium. Weitere werden folgen, wobei die c't-Redaktion auch auf Unterstützung aus dem Leserkreis rechnet: Wer SuperTape auf einem Mikrocomputer anderen Fabrikats implementiert, ist eingeladen, seine Lösung (selbstverständlich gegen Honorar) in c't vorzustellen.

Übrigens eignet sich SuperTape, wie prinzipiell jedes Kassetten-Aufzeichnungsverfahren, auch für die direkte Datenübertragung von Rechner zu Rechner. Man benötigt dazu lediglich ein Überspielkabel mit 'gekreuzten' Sende- und Empfangsleitungen. SuperTape dürfte also auch für die Besitzer von Floppy-Disk-Systemen interessant sein, die unter dem Format-Wirrwar der Hersteller nicht minder leiden.

Ursprünglich wurde das SuperTape-Verfahren für den COBOLD-Computer unserer Schwesterzeitschrift elrad entwickelt. Schon bald stellte sich heraus, daß dieses Verfahren (bei der zunächst gewählten Übertragungsgeschwindigkeit von 4800 Baud) außerordentlich zuverlässig arbeitet. Der COBOLD bietet die komfortable Möglichkeit, auf einen Timer im RIOT-Baustein zurückzugreifen, wodurch sich die Programmierung der SuperTape-Routinen recht einfach gestaltet. Als sich aber zeigte, daß diese auch mit einem Z80-Prozessor ohne Timer zu verwirklichen sind, lag die Weiterentwicklung zu einem universellen Standard auf der Hand. Besonders eindrucksvoll ist dabei der minimale Hardware-Aufwand, der sich bei vorhandenen Portleitungen auf nur einen Schmitt-Trigger beschränkt. Dieser ist bei allen Rechnern schon vorhanden, die mit zum Teil sehr aufwendigen — aber weniger leistungsfähigen — Kassetteninterfaces ausgestattet sind.

Eine wichtige Voraussetzung für SuperTape besteht darin, daß sämtliche Interrupts abschaltbar sein müssen. In den meisten Fällen ist diese Voraussetzung erfüllt, denn auch die anderen Kassettenverfahren müssen 'Interrupt-frei' arbeiten.

Auch eine andere Eigenschaft hat SuperTape mit den herkömmlichen Verfahren gemeinsam: Es ist empfindlich gegen Drop-outs. Darum sollten nur gute Bänder Verwendung finden. Außerdem darf natürlich der Kassettenrecorder kein übermäßiges Brummen oder Spikes vom Antriebsmotor einkoppeln.

Um dem Anspruch — SuperTape für jeden Rechner — gerechtzuwerden, durfte die Standard-Baudrate nicht zu hoch gewählt werden. Zudem sollen ja auch preiswerte Rekorder verwendet werden können. Bei der Standard-Übertragungsrate (3600 Baud) ist dies gewährleistet, denn die höchste zu übertragende Frequenz entspricht der Baudrate! Wer einen etwas besseren Recorder und einen nicht allzu 'lahmen' Mikrocomputer besitzt, kann praktisch ohne Einbußen an Übertragungssicherheit auf 7200 Baud umschalten. Allerdings dürfte sich diese Baudrate für den Software-Tausch nicht so gut eignen.

Die Technik

SuperTape arbeitet, wie die meisten Magnetband-Aufzeichnungsverfahren, mit Signalen verschiedener Frequenz. Eine logische 0 wird mit einer Schwingungsperiode der höheren, eine logische 1 mit einer halben Schwingungsperiode der niedrigeren Frequenz übertragen. Daraus resultiert eine Baudrate (Bit/sec), die gleich der höchsten Signalfrequenz ist. Bild 1 zeigt, wie das ausgesendete Bitmuster des Bytes D4h aussieht.

Beim Lesen von der Kassette (Bild 2) wird ein sehr einfaches Prinzip angewendet: Die Leseroutine ermittelt den Beginn der Übertragung eines Bits (t1) und testet über einen entsprechenden Port das Ausgangssignal am Schmitt-Trigger zum Zeitpunkt t2. Dann vergleicht sie den zum Zeitpunkt t2 ermittelten logischen Pegel mit dem

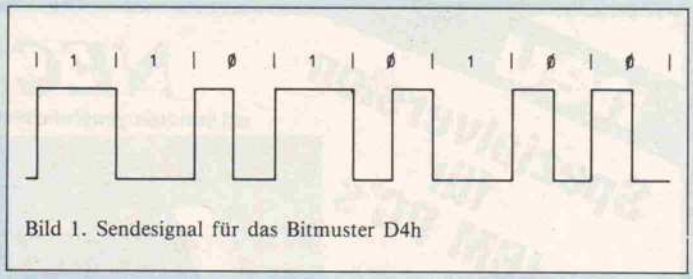


Bild 1. Sendesignal für das Bitmuster D4h

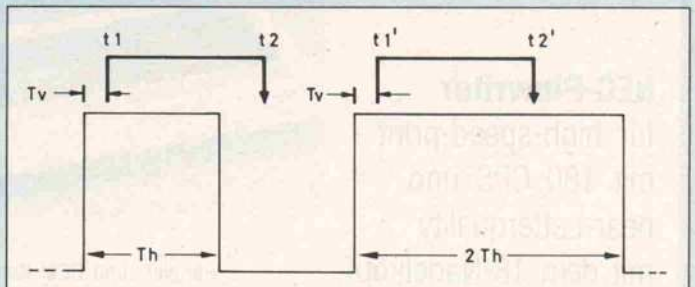


Bild 2. Lesevorgang: links für 0; rechts für 1. Die Verzögerungszeit Tv ist stark übertrieben

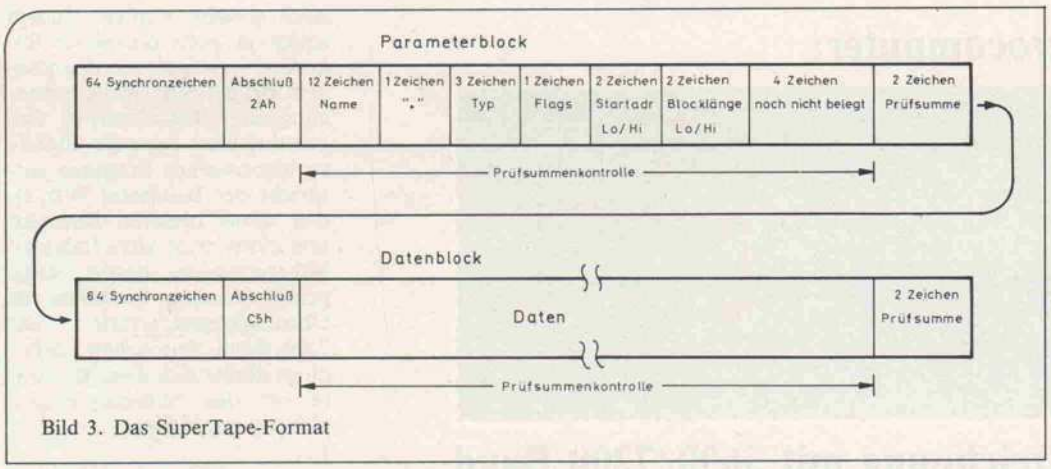


Bild 3. Das SuperTape-Format

gespeicherten vom Zeitpunkt t1. Sind beide gleich, so entspricht das einer 1; sind sie verschieden, erkennt sie eine 0.

Der optimale Zeitpunkt für t2 ergibt sich aus der Halbperiodendauer (Th) der höheren Schwingung und der mittleren Verzögerung (Tv) bei der Erkennung der Flanke zu

$$t2—t1 = 4/3 * Th — Tv .$$

Er liegt also nicht in der Mitte der zweiten Periodenhälfte, sondern ist ein wenig zu kürzeren Zeiten hin verschoben. Das hängt damit zusammen, daß Geschwindigkeitsschwankungen des Kassettenrecorders relative Abweichungen (in Prozent) sind. Deshalb ist der Prüfzeitraum t2—t1 so gewählt, daß sich im Idealfall (Tv = 0) eine Geschwindigkeitstoleranz von ±33% ergibt. Falls Ihr Kassettendeck tatsächlich Abweichungen in dieser Größenordnung aufweist, sollten Sie vielleicht an die Anschaffung eines neuen Gerätes denken. Der Praktiker kann den optimalen Prüfzeitraum natürlich auch experimentell ermitteln, indem er die Über- und Untergrenzen für korrektes Lesen feststellt und dazwischen entsprechend mittelt.

Gegenüber Störimpulsen, die dem Signal vom Band überlagert sind und zu Mehrfach-Triggerungen führen, ist das Verfahren unempfindlich. Es kommt lediglich darauf an, daß zum Zeitpunkt t2 ein definierter Pegel vorliegt.

Das Format

Die gesamte Datenübertragung findet bei SuperTape in zwei Blöcken statt (Bild 3). Zuerst wird ein Parameterblock gesendet, gefolgt von dem Datenblock. Der Parameterblock wird grundsätzlich mit 3600 Baud übertragen. Er beginnt mit 64 Synchronzeichen 16h und der Synchron-Abschlußmarke 2Ah. Dann folgen der Name (maximal 12 Zeichen), ein Punkt und drei Zeichen zur Kennzeichnung des Typs der übertragenen Daten (beispielsweise 'BAS' für ein BASIC-Programm).

Üblicherweise sind diese Zeichen ASCII-codiert. Das wird in dem anschließend übertragenen Flagregister durch eine Null in Flag 6 gekennzeichnet. Flag 7 gibt die Baudrate an, mit der der Datenblock übertragen wird; eine 1 steht für 7200 Baud. Die übrigen fünf Bits sind noch nicht belegt.

Es folgen die Startadresse und die Blocklänge des Datenblocks. Wie üblich, kommt auch hier das niederwertige Byte vor dem höherwertigen. Um für die Zukunft noch einige Erweiterungsmöglichkeiten offenzuhalten (z. B. eine Autostart-Adresse), werden danach vier Bytes, sozusagen als Platzhalter, gesendet. Zum Schluß folgen die beiden Bytes der Prüfsumme — ebenfalls Low vor High —, die bei dem Sendevorgang berechnet werden. Die Prüfsumme stellt die 16-Bit-Summe sämtlicher übertragenen Einsen dar. Die Berechnung beginnt hinter der Synchronabschlußmarke und endet vor den Prüfsummen-Bytes.

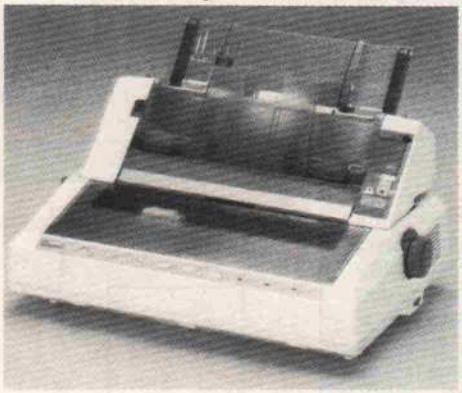
Der Datenblock beginnt gleichfalls mit den 64 Synchronzeichen 16h, die aber diesmal die Abschlußmarke C5h haben. Dann folgen die auszugebenen Daten und zum Schluß die Prüfsumme über den Datenbereich.

Beim Einlesen sind einige Optionen vorgesehen. Der gesuchte Name wird mit dem gelesenen verglichen. Ein '?' im Suchnamen bedeutet, daß das an gleicher Stelle im Lesenamen stehende Zeichen ignoriert wird. Ein '*' steht stellvertretend für den ganzen Rest des Namens.

Auch der Typ kann mit '?' oder '*' maskiert werden. Deshalb dürfen diese Zeichen genauso wie der '.' nicht Bestandteil von Name/Typ sein. SuperTape bietet auch die Möglichkeit, die gelesene Startadresse zu überschreiben, so daß der gesamte Datenblock in einen anderen Speicherbereich geladen wird.

neu
Spezialversion für IBM PC's

NEC
mit dem außergewöhnlichen Komfort



NEC-Typenkorb-Drucker Spinwriter
mit 20, 35 und 55 CPS für Briefqualität
— Ein-/Doppelschachtzuführung
— Einzelblattzuführung
— Umschlagszuführungsadapter
— Frontzuführung
— Schubtraktor
— Bi-direktionaler Traktor

NEC-Pinwriter
für high-speed-print mit 180 CPS und near-Letterquality mit dem 18-Nadelkopf.

Händler- und OEM-Konditionen auf Anfrage.

NEC-HAUPTHÄNDLER:
SYSDAT GmbH
Brabanterstr. 38—40
5000 Köln 1
Tel. 02 21/23 54 44
Telex 8 882 686

DR. DOHRENBERG
Bayreuther Str. 3
1000 Berlin 30
Tel. 0 30/2 13 80 43-45
Telex 01 84 860

SEH COMPUTER-PERIPHERIE GERÄTE GmbH
Maybachstr. 17
6450 Hanau 7
Tel. 0 61 81/6 20 57
Telex 4 15 508

SCHWIND DATENTECHNIK GmbH
Maria-Eich-Str. 45
8000 München 60
Tel. 0 89/8 34 97 16
Telex 5 213 097

SuperTape mit dem Z80

am Beispiel des ZX81/16K

Andreas Stiller

Auch mit dem einfachsten aller Z80-Rechner kann das SuperTape-Verfahren ohne jeglichen Hardwarezusatz verwirklicht werden. Die oftmals belächelte 'Schachtel' bewältigt dabei nicht nur problemlos die Übertragungsrate von 3600 Baud, sondern kann mit geeigneten Recordern bis zu 7200 Baud erreichen. Außerdem können die Programme und Daten des ZX81 problemlos von seinem größeren Bruder, dem Spectrum, gelesen werden, wenn auch dieser SuperTape-fähig ist. Am Ende dieses Beitrags schließen sich Hinweise für die SuperTape-Implementation auf anderen Z80-Rechnern an.

Da das gesamte Programm etwas zu umfangreich für eine Heftnummer ist (wer will schon 1000 Bytes auf einmal eintippen?), wurde es in zwei Teile zerlegt. Der erste Teil stellt die reinen SuperTape-Routinen vor, der zweite behandelt die komfortable Ein-/Ausgabe und die Kaltstartmöglichkeit.

Für die Standardrate von 3600 Baud eignet sich jeder 'Wald- und Wiesenrekorder', allerdings sollten Sie ein gutes Bandmaterial verwenden. Die Ansteuerung des Rekorders erfolgt wie gewohnt über die MIC- beziehungsweise EAR-Buchse. Das gesamte SuperTape-Programm ist vollständig in Maschinensprache geschrieben und beansprucht rund 1K. Damit verlieren Sie zwar etwas Speicherplatz, aber wenn Ihre Programme ungefähr zehnmal schneller als bisher zu retten und zu laden sind, werden Sie dieses Speicheropfer sicherlich gerne bringen. Hinzu kommt, daß Sie sicherlich viel öfter Zwischenspeicherungen bei der Programmentwicklung vornehmen werden, weil die nervtötenden Wartezeiten auf ein erträgliches Maß reduziert sind. Jeder, der sich schon mal nach einem Systemcrash die Haare gerauft hat, weil die letzte Zwischenspeicherung schon einge Arbeitsstunden zurücklag, wird das zu schätzen wissen.

Um das SuperTape-Verfahren mit einem Z80-Prozessor ohne Timer zu verwirklichen, muß man die genaue Taktfrequenz des Rechners kennen und die Laufzeit aller Befehle berücksichtigen. Wieviel Takte jeder Befehl braucht, läßt sich dem technischen Manual des Herstellers Zilog oder der Sekundärliteratur (zum Beispiel Rodnay Zaks: Programmierung des Z80) entnehmen. Wichtig ist dabei, daß die verwendeten Speicher nicht noch zusätzliche WAIT-Zyklen einstreuen, wodurch das Timing ziemlich durcheinander geriete. Glücklicherweise trifft das beim ZX81 mit handelsüblichen Speichern nicht zu. Auch der NMI-Generator läßt sich abschalten, ist also ganz im Gegensatz zu seinem Namen doch 'maskierbar'. Als Befehl dazu dient:

```
OUT ØFDH,A
```

Dieser Befehl taucht also auch in der im Programm verwendeten ROM-Routine 02E7H auf, die in eine spezielle Art des FAST-Mode, den wir hier als 'kurzfristigen FAST-Mode' bezeichnen, umschaltet.

Die Taktfrequenz des ZX81 liegt bei rund 3,25 MHz; sie ist leider nicht quarzstabilisiert und kann demnach auch etwas davon abweichen. Beim ZX des Autors ergab sich eine ausgemessene Taktfrequenz von 3,223 MHz. SuperTape ist jedoch gegenüber Zeitabweichungen so großzügig, daß sich hieraus noch keinerlei Probleme für die sichere Datenübertragung ergeben. Im Testbetrieb zeigte sich, daß sogar Baudraten von 9600 Baud erreichbar sind, allerdings auf einem Spulengerät mit höherer Bandgeschwindigkeit. Üblicherweise sollte man sich ruhig auf eine Baudrate von 3600 beschränken, damit die Aufzeichnungen auch von langsameren SuperTape-fähigen Rechnern gelesen werden können.

Aller Anfang ...

Da die meisten ZX-Benutzer wohl keinen Assembler zur Verfügung haben, muß das Programm Byte für Byte mit einer kleinen Hilfsroutine eingegeben werden. Das schöne Assembler-Listing der SuperTape-Routinen ist leider auch nicht mit dem ZX erstellt worden, sondern mit einem CP/M-Rechner. Wenn auch dieser erst einmal SuperTape-fähig ist, könnte man assemblierte Programme direkt auf den Sinclair überspielen.

Als Speicherbereich ist für SuperTape der Platz ab 7C00H (31744) vorgesehen. Dieser Bereich wird vor der Eingabe und vor jedem LOAD mit

```
POKE 16389,124
NEW
```

geschützt. Alsdann können Sie folgendes Hexbyte-Eingabeprogramm eintippen:

```
1000 LET P=16508
1100 FAST
1200 LET P=P+6
1300 LET X=PEEK P
1400 IF X=227 THEN STOP
1500 IF X=118 THEN
     GOTO 1200
1550 IF X=20 THEN
     GOTO 2000
1600 POKE R,16*X +
     PEEK (P+1) —476
1700 LET P=P+2
1800 LET R=R+1
1900 GOTO 1300
2000 LET R= 4096* PEEK
   (P+1)+ 256* PEEK(P+2)
   + 16 * PEEK (P+3) +
   PEEK (P+4) — 122332
2100 LET P=P+5
2200 GOTO 1300
```

Mit diesem kleinen Programm können Sie schon relativ komfortabel in Maschinensprache programmieren: der Hexcode wird in REM-Statements beliebiger Länge gespeichert und kann somit auch relativ einfach mit den üblichen Editierbefehlen korrigiert werden. Das Programm erkennt das Ende eines REM-Befehls und springt dann zum ersten Hexcode des nächsten REMs — bis irgendwann ein STOP (Code 227) auftaucht. Falls als erstes Zeichen im REM-Statement ein '=' auftritt, interpretiert der 'Mikro-Assembler' die folgenden vier Bytes als hexadezimale Origin-Anweisung, also als Startadresse für den folgenden Hexcode.

Denken Sie daran, daß im Hexcode kein 'H' mehr zur Kennzeichnung der Zahlenbasis auftreten darf!

Das SuperTape-Programm beginnt also entsprechend dem Assembler-Listing (Hexcode in der zweiten Spalte) mit:

```
28 REM = 7DA8
29 REM 21DF7F
30 REM 011900
31 REM 3E2A
```

und endet mit: REM STOP.

Verwenden Sie wirklich den 'STOP'-Befehl, also Shift A (und nicht etwa ausgeschrieben). In eine Zeile können Sie natürlich auch mehrere Maschinensprachenbefehle unterbringen, doch dürfte die dem Listing folgende Aufteilung am übersichtlichsten bleiben. Wenn Sie alle Befehle eingetippt und kontrolliert haben, sollten Sie das Programm — vorerst noch im Sinclair-Verfahren — auf Band retten. Mit

```
GOTO 1000
```

beginnt dann der Mikroassembler seine Arbeit und erzeugt den Maschinencode.

Die Puffer

SuperTape benötigt zwei Pufferbereiche:

den SV-Puffer
von 7FCEh bis 7FE6h,

in den die gewünschten Werte für Save und Load geschrieben werden,

den LD-Puffer
von 7FE7h bis 7FFFh,

für die vom Band gelesenen Werte.

Die Pufferbelegung:

SV	LD	
7FCEh	7FE7h	Name
7FDAh	7FF3h	Punkt
7FDBh	7FF4h	Typ
7FDEh	7FF7h	Flags
7FDFh	7FF8h	Startadr

7FE1h	7FFAh	Länge
7FE3h	7FFCh	frei
7FE6h	7FFFh	Ende

Die Puffer sind gemäß dem SuperTape-Standard jeweils 25 Bytes lang. Zusätzlich zu den Puffern braucht der ZX81 noch zwei Register:

Flag 0: gesetzt für Save
Flag 1: gesetzt Verify
Flag 2: gesetzt neue Startadr.
Flag 6: gesetzt nicht ASCII
Flag 7: gesetzt 7200 Bd für Daten

7FCDh (32717) Baudratenregister
muß 00h für den Parameterblock sein und wird für 7200 bd dann 80h

Das ZFlag — und das Baudratenregister, sowie de SV-Puffer müssen Sie vorerst noch 'von Hand', das heißt mittels POKE, oder besser mit Hilfe des Mikroassemblers beschicken. Für die nächste Ausgabe ist vorgesehen, dafür geeignete und komfortable Ein-/Ausgaberoutinen zu veröffentlichen. Für das mitgesendete SV-Flagregister sind bislang nur die Flags 6 und 7 interessant, die denen der ZFlags entsprechen.

Als Beispiel sollen die ersten 100h Bytes des ROM auf Band gerettet werden: Im ZX81-Code, mit 3600 Baud. Der Name sei Test und vom Typ Hex.

```
1 REM =7FCC
2 REM 4100
3 REM 392A38390000000
  0000000001B2D2A3D
4 REM 40000000010000
5 REM STOP
```

Mit GOTO 1000 wird der Puffer dann geladen. Lassen Sie den FAST-Mode bestehen. Nun endlich können Sie mit

PRINT USR 32168

das erste Mal SuperTape in Aktion 'sehen' und auch hören, sofern Sie schon einen Kassettenrekorder, auf Aufnahme geschaltet, an der MIC-Buchse angeschlossen haben.

Zum Laden wird eine andere Startadresse gewählt, so daß die ROM-Seite in den RAM-Speicher (ab 6000 h = 24576) kopiert wird. Außerdem sollten Sie noch zusätzlich eine kleine Routine verwenden, die eventuelle Fehlermeldungen ausgibt. Der Fehlercode wird von LDSUPER im A-Register gespeichert, und zwar:

00h = 0 = ' ' fehlerfrei
27h = 39 = 'B' falscher Name oder Typ
2Bh = 43 = 'F' Fehler bei der Prüfsumme
12h = 18 = '>' Fehler bei Verify

Wenn Sie diesen Code in das BC-Register bringen, erhalten Sie mit dem USR-Befehl den dezimalen Wert. Diese Aufgabe erledigt folgendes Maschinenprogramm:

```
CALL LDSUPER
LD   C,A
LD   B,0h
RET
```

In unserem Mikroassembler sieht das so aus:

```
1 REM =7FCC
2 REM 4400
3 REM 392A38390000000
  0000000001B2D2A3D
4 REM 400060
5 REM =7C00
6 REM CD547E
7 REM 4F
8 REM 0600
9 REM C9
10 REM STOP
```

Die Zusatzroutine beginnt demnach bei 7C00h. Sie wird nach GOTO 1000 — weiterhin im normalen FAST-Mode — mit PRINT USR 31744 gestartet. Daraufhin wartet der Rechner auf Signale am EAR-Eingang, kann also jetzt den Datenblock laden. Wenn auf dem Schirm zum Schluß eine Null erscheint, dann ist der Ladevorgang aller Wahrscheinlichkeit nach korrekt erfolgt, wovon Sie sich mit PEEKs ab 24576 überzeugen können.

Wenn Sie beim Laden einzelne Buchstaben des Namens oder des Typs durch ein '?' (Code 0Fh) ersetzen, wird das eingelesene Zeichen an dieser Stelle übersprungen. Ein '*' (Code 17h) überliest sogar bis zum jeweiligen Ende des Strings.

Piepshow

Die Möglichkeit, ein Ausgangssignal zu erzeugen, ist beim ZX81 durch zwei I/O-Befehle gegeben:

```
OUT FFh,A   ;einschalten
IN  FEh,A   ;ausschalten
```

Mit dem OUT-Befehl wird das Signal an der MIC-Buchse im Zeilenfrequenzrhythmus kurzzeitig wieder abgeschaltet (16 Takte lang). Allerdings wird dieser kurze Puls durch das eingebaute Bandpaßfilter weitgehend abgefangen, so daß sich bei den benutzten Baudraten noch keine Probleme ergeben. Um den 'OUT FFh,A'-Befehl überhaupt ausführen zu können, muß zuvor mit 'OUT FDh,A' der NMI-Generator umgeschaltet werden. Wenn der Akkumulator beim IN-Befehl auf 7Fh steht, wird nicht nur das Ausgangssignal abgeschaltet, sondern mit diesem Befehl auch die untere rechte Zeile der Tastatur gelesen. Damit können Sie also zum Beispiel auf BREAK testen (Bit 0 = 0). Darüber hinaus ist dieser Befehl auch für das Einlesen der Signale vom Band verantwortlich: bei erkanntem Signal ergibt sich Bit 7 zu eins.

Taktgefühl

Die Zahl der notwendigen Takte ergibt sich aus der halben Periodendauer der höheren Schwingung. Bei 7200 Bd beträgt sie etwa 69 Mikrosekunden, entspricht also 224 Takten. In Bild 1 ist das Timing-Diagramm für die SVBYTE-Routine wiedergegeben. Der Balken bedeutet die Inversion des Ausgangssignals. Zwischen zwei Inversionen liegen bei Ausgabe einer Null 147 + SVWAIT Takte, und bei einer Eins das Doppelte. Folglich muß die Verzögerungsroutine SVWAIT insgesamt 224 — 147 = 77 Takte bei 7200 und 448 — 147 = 301 Takte bei 3600 Baud 'vernichten'. Die INVERT-Routine ist beim ZX81 etwas umständlich, da ja zwei verschiedene Ports angesprochen werden müssen. Andere Z80-Rechner, die dafür nur ein Port verwenden, können gerade hier einige Takte sparen.

Für das Bereitstellen der auszugebenden Daten stehen den aufrufenden Routinen nur 33 Takte zur Verfügung. Um das zu gewährleisten, wird beim letzten Byte in der SVBLOC-Routine die Schleifenkontrolle eingespart. Deshalb muß der Datenblock mindestens zwei Bytes lang sein — was wahrlich keine Einschränkung darstellt.

Bild 2 gibt das Timing-Diagramm der Laderoutine LDBYTE wieder. Die Testzeit für SuperTape ist am günstigsten für:

halbe Periode bei Null * 4/3 — mittlere Flankenerkennungszeit.

Der letzte Anteil ergibt sich in LDBYTE zu ungefähr 17 bis 18 Takten. Als Testzeit errechnet sich dann für

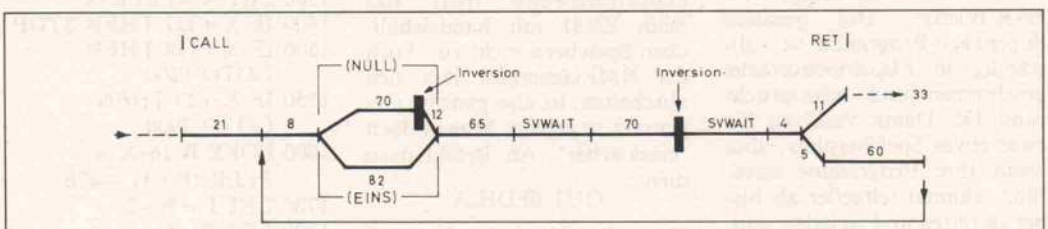


Bild 1. Timing für SVBYTE: 147T + SVWAIT zwischen zwei Inversionen

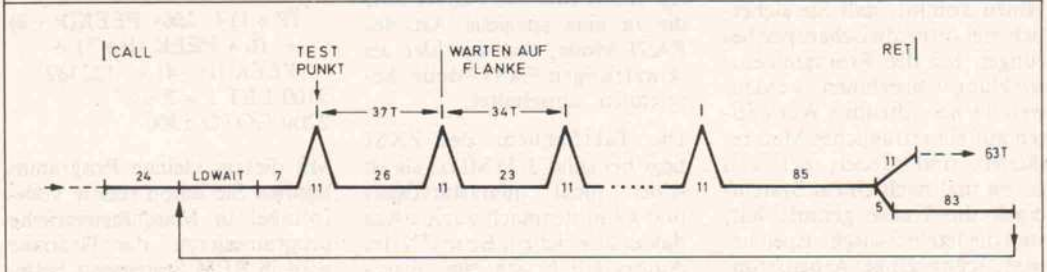


Bild 2. Timing für LDBYTE: 201T + LDWAIT für den Prüfzeitraum

7200 Baud : 280 Takte
3600 Baud : 580 Takte

Der Eigenverbrauch der Routine ist 201 Takte, so daß CALL LDWAIT noch 79 Takte bei 7200 beziehungsweise 379 Takte bei 3600 Baud abwarten muß. Für die Verarbeitung der eingelesenen Daten verbleiben den aufrufenden Routinen dann noch 63 Takte.

Die Hierarchie

Bild 3 zeigt, in welcher Weise die Save-Routinen untergeordnet sind. Das Hauptprogramm SVSUPER gibt erst den SV-Puffer an das Unterprogramm SVBLOC und dann die auszugebenden Daten. SVBLOC übergibt dann die Synchronbytes, die Abschlußmarke, dann die Daten und schließlich die Prüfsumme und das Abschlußbyte an das Unterprogramm SVBYTE. Hier wird dann jedes Byte Bit für Bit mit den INVERT-Routinen in die entsprechenden SuperTape-Pulse übertragen. SVWAIT sorgt dabei für die richtige Baudrate.

Ähnlich sieht die Ordnung bei den Ladeprogrammen aus (Bild 4). Das Hauptprogramm LDSUPER läßt zuerst von LBLOC den Parameterblock laden, dann wird überprüft, ob Name und Typ stimmen. Je nach gesetztem Flag wird eine ASCII-ZX81-Code-Umwandlung durchgeführt. Dann kontrolliert das Programm, ob eine neue Startadresse gesetzt werden soll und führt schließlich entweder die Lade-, oder die Verify-Routine aus.

LDBLOC ruft zunächst erst einmal die Synchronisier-Routine LDSYNC auf, die versucht, auf die Synchronbytes einzurasten und das richtige Abschlußbyte kontrolliert. Dann werden die Daten mittels LDBYTE geladen und schließlich zu LDTEST gesprungen, daß die Prüfsummen vergleicht. LDVER sieht genauso wie LDBLOC aus, nur daß es die Bytes mit denen im Speicher vergleicht, statt sie zu laden.

LDBYTE macht dann das Umgekehrte wie SVBYTE, indem es den Bandinhalt Bit für Bit einliest und diese ins E-Register schiebt. In diesem Fall ist LDWAIT für die richtige Baudrate verantwortlich.

Die einzelnen Routinen sind bis auf SVBYTE und LDBYTE so einfach strukturiert, daß sie wohl ohne Schwierigkeiten anhand des Listings verstanden werden können. Als einziger zeitsparender Trick tritt dabei die Kombination

```
CPI
JP PE
```

in den Blockübertragungsschleifen von SVBLOC (bzw. LDBLOC)auf. Der Vergleich interessiert hierbei gar nicht, sondern die Tatsache, daß CPI das Adreßregister HL inkrementiert, das Zählregister BC dekrementiert und das PO/PE-Flag setzt. Dieses steht auf PE, solange BC noch nicht Null ist. Mit diesem Trick kommt man für die Schleifenüberwachung mit 26 Takten aus.

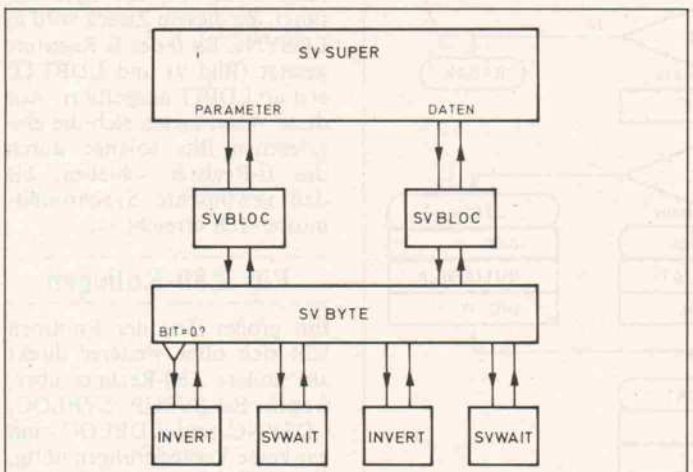


Bild 3. Die SAVE-Routinen

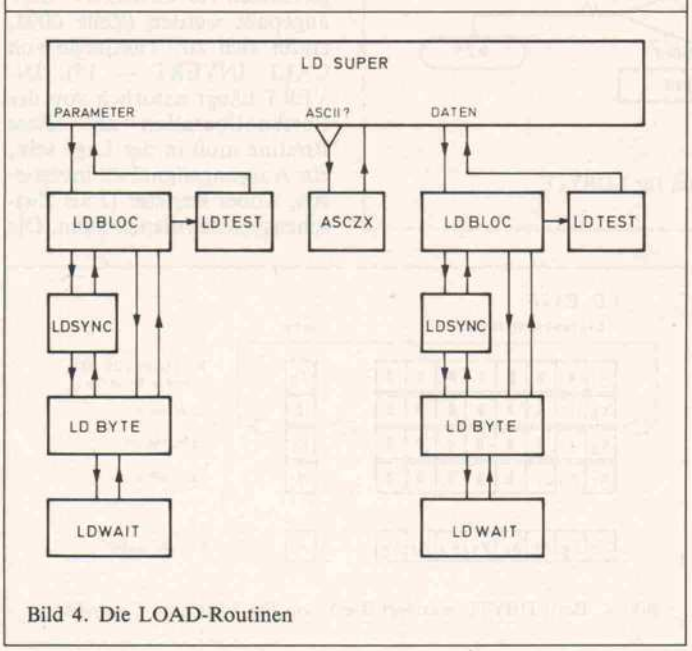


Bild 4. Die LOAD-Routinen

Das Schreiben ...

Die beiden Routinen SVBYTE und LDBYTE, die letztendlich die Bytes auf das Band bringen, beziehungsweise vom Band lesen, enthalten ebenfalls einen zeitsparenden Trick: ein Register übernimmt zusammen mit dem Carry-Flag sowohl die Bit-Ein-/Ausgabe als auch die Schleifenkontrolle. Wie das geschieht, zeigen für SVBYTE das Flußdiagramm (Bild 5) und die Registerdarstellung (Bild 7).

SCF setzt das Endflag, das nach und nach mit RR E durch das Ausgaberegister geschoben wird, wobei gleichzeitig das auszugebende Bit (also das niederwertige Bit zuerst) in das Carry-Flag gelangt. Hier kann es nun abgefragt und ausgewertet werden. Zum Schluß sorgt XOR A dafür, daß bei den nächsten RR E-Befehlen von links nur Nullen nachrücken. Auf diese Weise werden nacheinander alle Bits ins Carry geschoben, bis das E-Register auf Eins steht. DEC E fragt ab, ob diese Eins schon erreicht ist. Falls ja, erfolgt der Rücksprung ins aufrufende Programm. Im anderen Fall wird mit INC E das Ausgaberegister wieder restauriert und das nächste Bit verarbeitet.

Die Auswertung

Eine Null im Carry bedingt einen Sprung ins Unterprogramm INVERT, das das Ausgangssignal invertiert. Eine Eins läßt diesen Sprung aus, inkrementiert dafür aber das Prüfsummenregister IY. Das folgende WAIT 55T bedeutet, daß der Rechner die nächsten 55 Takte nichts anderes tun soll, als nur die im Timingdiagramm (Bild 1) dargestellte Laufzeit von Call INVERT auszugleichen. Leider gibt es beim Z80 den WAIT-Befehl nicht, sondern es müssen dafür Dummy-Befehle gefunden werden, die genau diese Taktanzahl verbrauchen, ohne dabei wichtige Register zu verändern. Um diese Dummies zu kennzeichnen, sind sie im Listing als DEFB aufgeführt.

Daraufhin wartet das Programm unabhängig von dem Ausgangsbit 65 Takte, ruft die Routine SVWAIT auf, die für die Baudrate verantwortlich ist, invertiert das Ausgangssignal und ruft nochmals SVWAIT für die nächste Periodenhälfte auf.

... und das Lesen

Die Funktion von LDBYTE verdeutlicht das Flußdiagramm (Bild 6) und die Registerdarstellung (Bild 8). Hier wird eine Eins von links nach rechts durch das anfangs leere E-Register geschoben, bis sie ins Carry gelangt, woraufhin der Rücksprung erfolgt. Bei dieser Schiebeaktion gelangen die eingelesenen Bits aus dem Carry ins E-Register.

Die Routine testet zunächst mit IN A,(FEh) den Zustand an der EAR-Buchse und vergleicht ihn mit dem im Register D gespeicherten. Um Zeit zu sparen, werden die gesetzten Flags auf den Stack gerettet und sofort die Flankenabfrage begonnen. Register D erhält dabei den eingelesenen Wert, und die FLWAIT-Schleife wird solange durchlaufen, bis sich ein geänderter Wert am Eingang ergibt. Dieser Wert kommt nun wieder in den Zustandsspeicher D. Ein

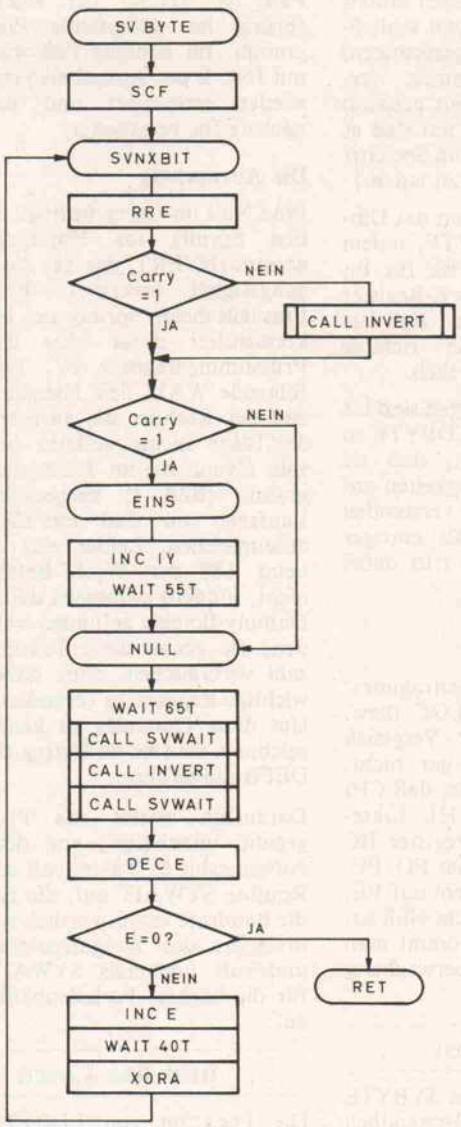


Bild 5. Flußdiagramm für SVBYTE

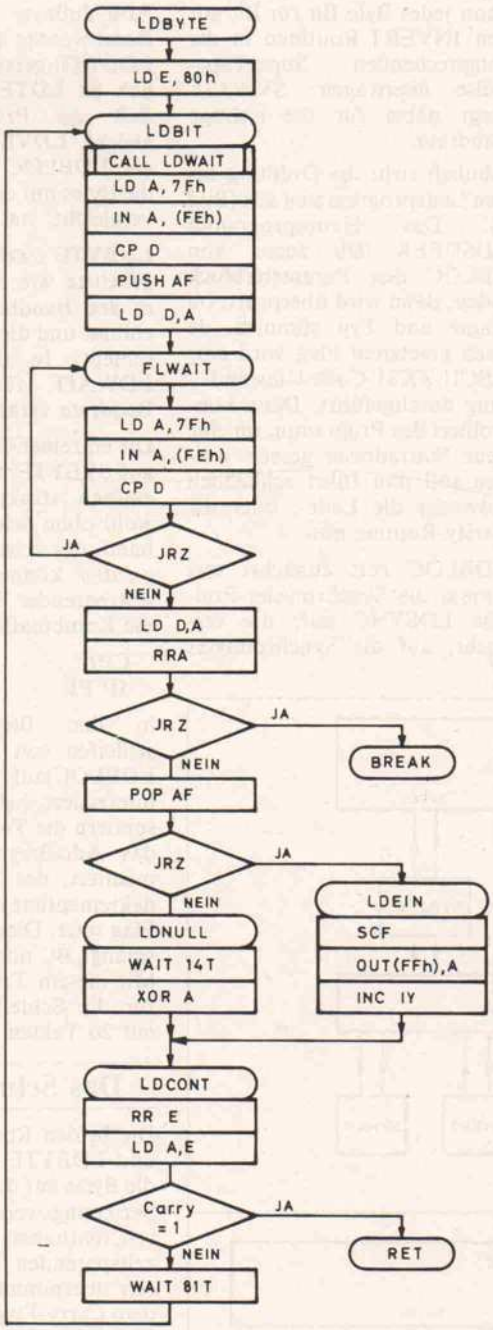


Bild 6. Flußdiagramm für LDBYTE

geänderter Wert kann sich aber auch durch BREAK ergeben haben, was mit RRA und JR NC,BREAK kontrolliert wird.

Jetzt ist genug Zeit da, sich an die Auswertung des Vergleichs zu begeben. Das Flagregister wird wieder vom Stack geholt, und das Carry-Flag wird entsprechend gesetzt:

Zwei gleiche Zustände ergeben eine Eins und bewirken eine Erhöhung der Prüfsumme in IY und eine Bildschirmausgabe (OUT (FFh),A);

ungleiche Zustände haben Carry-Flag = Ø zur Folge.

Damit ist das Eingangssignal als Bit interpretiert worden, welches nun nach dem oben beschriebenen Mechanismus in das E-Register wandert.

Für die Synchronisierung ist das bitweise Einlesen erforderlich, da der Rechner ja noch nicht weiß, wo ein Byte anfängt. Zu diesem Zweck wird in LDSYNC Bit 0 des E-Registers gesetzt (Bild 9) und LDBYTE erst ab LDBIT ausgeführt. Auf diese Weise lassen sich die eingelesenen Bits solange durch das E-Register schieben, bis daß gewünschte Synchronbitmuster 16h erreicht ist.

Für Z80-Kollegen

Ein großer Teil der Routinen läßt sich ohne weiteres direkt auf andere Z80-Rechner übertragen. Bei SVSUP, SVBLOC, LDSYNC und LDBLOC sind gar keine Veränderungen nötig, in SVBYTE muß die Ausgleichszeit für CALL INVERT angepaßt werden (Zeile 0093, ergibt sich zu: Taktlänge von CALL INVERT — 15). INVERT hängt natürlich von der Portkonfiguration ab. Diese Routine muß in der Lage sein, ein Ausgangssignal zu invertieren, wobei Register D als Zwischenspeicher dienen kann. Die

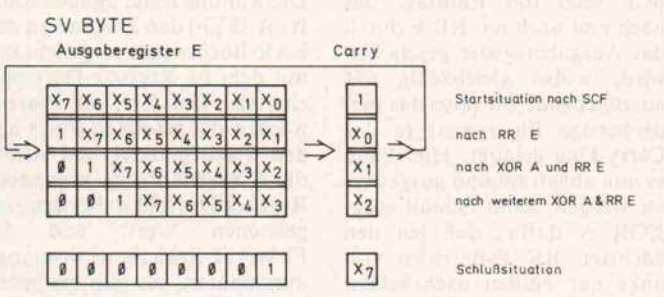


Bild 7. Bei SVBYTE wird eine 1 von Carry bis Bit 0 geschoben

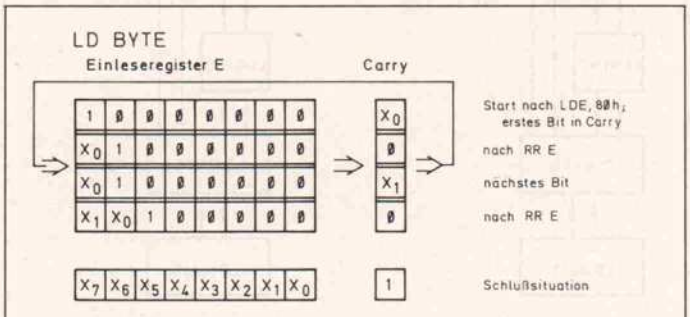


Bild 8. Bei LDBYTE wandert die 1 von Bit 7 bis zum Carry-Flag

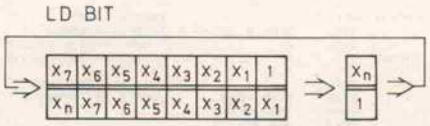


Bild 9. Für LDBIT muß Bit 0 gesetzt werden

Taktlänge muß in jedem Fall konstant bleiben. Achten Sie darauf, daß das Carry-Flag beim Rücksprung nicht gesetzt sein darf.

Die Routinen SVWAIT und LDWAIT sind die einzigen, die von der Taktfrequenz des Rechners abhängen. Die Berechnung der Taktlängen folgt analog der Beschreibung im Kapitel 'Taktgefühl'.

In LDSUPER ist nur die Fehlermeldung in Zeile 186 durch einen anderen Code (normalerweise hoffentlich ASCII) zu ersetzen. Zeilen 175, 183 und 184 können dann entfallen. Auch in LDVER und LDTEST sind nur Änderungen in den Fehlermeldungen nötig (Zeile 238 und 256). Auf die Codewandlungsroutinen ab Zeile 371 können Sie natürlich verzichten.

In LDBYTE müssen Sie über ein Port die Signalabfrage durchführen können. Falls auch ihr Rechner, ähnlich wie der ZX81, nicht über eine Warmstartmöglichkeit verfügt, sollten Sie hier auch eine BREAK-Abfrage einbeziehen. Der OUT-Befehl in Zeile 335 kann entfallen, woraufhin allerdings auch der WAIT-Befehl in Zeile 328 verschwinden muß und XOR A in XOR (HL) zu verwandeln ist. Mit Hilfe eines Assembler-Handbuches können Sie dann ein Timing-Diagramm ähnlich wie Bild 2 erstellen und die notwendigen Verzögerungen für LDWAIT bestimmen.

Vergessen Sie nicht, alle Interrupts abzuschalten, bevor Sie die SuperTape-Routinen benutzen!

```
                        0001
                        0002
                        0003            ;SUPER TAPE
                        0004            ;----------
        (7DA8)          0005            ORG     7DA8H
                        0006            ;----------
        (7FCC)          0007  ZFLAGR    EQU     7FCCH
        (7FCD)          0008  BDRATE    EQU     7FCDH
        (7FCE)          0009  SVNAME    EQU     7FCEH
        (7FDB)          0010  SVTYPE    EQU     7FDBH
        (7FDE)          0011  SVFLAGR   EQU     7FDEH
        (7FDF)          0012  SVSTART   EQU     7FDFH
        (7FE1)          0013  SVLEN     EQU     7FE1H
        (7FE7)          0014  LDNAME    EQU     7FE7H
        (7FF4)          0015  LDTYPE    EQU     7FF4H
        (7FF7)          0016  LDFLAGR   EQU     7FF7H
        (7FF8)          0017  LDSTART   EQU     7FF8H
        (7FFA)          0018  LDLEN     EQU     7FFAH
        (7FFF)          0019  LDEND     EQU     7FFFH
        (7FCB)          0020  TABEND    EQU     7FCBH
                        0021            ;
        (002A)          0022  SYNCPAR   EQU     2AH
        (00C5)          0023  SYNCDAT   EQU     0C5H
        (0016)          0024  SYNCHAR   EQU     16H
                        0025  ;
                        0026  ;-------------------------------------------------
                        0027  ;-----  *****   SVSUP: SUPERSAVE    **** ----------
                        0028  ;-------------------------------------------------
7DA8  21CE7F            0029  SVSUP     LD      HL,SVNAME       ;SENDET PARAMETERBLOCK
7DAB  011900            0030            LD      BC,0019H        ;       "
7DAE  3E2A              0031            LD      A,SYNCPAR
7DB0  CDCA7D            0032            CALL    SVBLOC          ;       "
                        0033  ;                                 ----------------------
7DB3  3ADE7F            0034            LD      A,(SVFLAGR)     ;SETZT BAUDRATE
7DB6  32CD7F            0035            LD      (BDRATE),A
                        0036  ;                                 ----------------------
7DB9  2ADF7F            0037            LD      HL,(SVSTART)    ;SENDET DATENBLOCK
7DBC  ED4BE17F          0038            LD      BC,(SVLEN)      ;       "
7DC0  3EC5              0039            LD      A,SYNCDAT
7DC2  CDCA7D            0040            CALL    SVBLOC          ;       "
7DC5  FD210040          0041            LD      IY,4000H
7DC9  C9                0042            RET
                        0043  ;
                        0044  ;-------------------------------------------------
                        0045  ;-----  *****   SVBLOC: SENDET BLOCK ***** ----------
                        0046  ;-------------------------------------------------
7DCA  1600              0047  SVBLOC    LD      D,00H           ;SETZT INVERT-REGISTER
7DCC  F5                0048            PUSH    AF              ;SYNC-ABSCHLUSS: STACK
7DCD  0B                0049            DEC     BC              ;DEKR. BLOCKLÄNGE
7DCE  C5                0050            PUSH    BC              ;BLOCKLÄNGE AUF STACK
                        0051  ;                                 ----------------------
7DCF  0640              0052            LD      B,40H           ;SENDE 64 SYNC-ZEICHEN
7DD1  1E16              0053  SVLOOP1   LD      E,SYNCHAR       ;LADE E MIT SYNC-BYTE
7DD3  A7A7D8            0054            DEFB    0A7H,0A7H,0D8H  ;WARTE 13 TAKTE
7DD6  CD077E            0055            CALL    SVBYTE
7DD9  10F6              0056            DJNZ    SVLOOP1
                        0057  ;                                 ----------------------
7DDB  C1                0058            POP     BC              ;HOLE LÄNGE VOM STACK
7DDC  F1                0059            POP     AF              ;HOLE SYNC-ABSCHLUSS
7DDD  5F                0060            LD      E,A             ;SENDE SYNC-ABSCHLUSS
7DDE  CD077E            0061            CALL    SVBYTE
                        0062  ;                                 ----------------------
7DE1  FD210000          0063            LD      IY,0000H        ;INIT PRÜFSUMMENZÄHLER
7DE5  A6D8              0064            DEFB    0A6H,0D8H       ;WARTE 11 TAKTE
7DE7  5E                0065  SVLOOP2   LD      E,(HL)          ;SENDE BYTE
7DE8  CD077E            0066            CALL    SVBYTE
7DEB  EDA1              0067            CPI
7DED  EAE77D          P 0068            JP      PE,SVLOOP2
7DF0  5E                0069            LD      E,(HL)
7DF1  CD077E            0070            CALL    SVBYTE
                        0071  ;                                 ----------------------
7DF4  C0                0072            RET     NZ              ;WARTE 5 TAKTE
7DF5  FDE5              0073            PUSH    IY              ;SENDE PRÜFSUMME:
7DF7  C1                0074            POP     BC
7DF8  59                0075            LD      E,C
7DF9  CD077E            0076            CALL    SVBYTE
7DFC  3435A6            0077            DEFB    34H,35H,0A6H    ;WARTE 29 TAKTE
7DFF  58                0078            LD      E,B
7E00  CD077E            0079            CALL    SVBYTE
7E03  E3E3              0080            DEFB    0E3H,0E3H       ;WARTE 50 TAKTE
7E05  A6D8              0081            DEFB    0A6H,0D8H       ;UND
                        0082                                    ;SENDE SCHLUSSBYTE
                        0083  ;
                        0084  ;-------------------------------------------------
                        0085  ;-----  *****   SVBYTE: RETTET BYTE AUF BAND *****------
                        0086  ;-------------------------------------------------
7E07  37                0087  SVBYTE    SCF                     ;SETZE ENDFLAG
7E08  CB18              0088  SVNXBIT   RR      E               ; AUSGABEBIT IN CARRY
7E0A  D42F7E            0089            CALL NC,INVERT          ; BIT=0? INV. AUSGANGS-
7E0D  3007              0090            JR NC,  NULL            ;SIGNAL, SPRINGE ZU NULL
                        0091  ;                                 ----------------------
7E0F  FD23              0092  EINS      INC     IY              ;INKR PRÜFSUMMENZAHLER
7E11  E3E3              0093            DEFB    0E3H,0E3H       ;WARTE 55 TAKTE
7E13  A6D8D8            0094            DEFB    0A6H,0D8H,0D8H  ;       "
                        0095  ;                                 ----------------------
7E16  E3E3              0096  NULL      DEFB    0E3H,0E3H       ;WARTE 65 TAKTE
7E18  E523E1            0097            DEFB    0E5H,23H,0E1H   ;       "
7E1B  CD447E            0098            CALL    SVWAIT          ;WARTE ENTSPR. BAUDRATE
7E1E  CD2F7E            0099            CALL    INVERT          ;INV. AUSGANGSSIGNAL
7E21  CD447E            0100            CALL    SVWAIT          ;WARTE ENTSPR. BAUDRATE
7E24  1D                0101            DEC     E               ;RETURN FALLS E=1:
7E25  C8                0102            RET     Z               ;ALLE 8 BIT AUSGEGEBEN
7E26  1C                0103            INC E
7E27  E5E1              0104            DEFB    0E5H,0E1H       ;WARTE 40 TAKTE
7E29  A6A6D8            0105            DEFB    0A6H,0A6H,0D8H
7E2C  AF                0106            XOR     A               ;SETZE CARRY AUF 0
7E2D  18D9              0107            JR      SVNXBIT
                        0108  ;
                        0109  ;-------------------------------------------------
                        0110  ;-----  *****   INVERT: INVERTIERT AUSGANGSSIGNAL ***
                        0111  ;-------------------------------------------------
7E2F  15                0112  INVERT    DEC     D               ;D=1? D:=0 UND PULS
7E30  2808              0113            JR Z,   PULSAUS         ;WIRD ABGESCHALTET
                        0114  ;
                        0115  ;-----  *****   PULSEINSCHALTUNG    *****------------
                        0116  ;
7E32  14                0117  PULSEIN   INC     D               ;D=0? D:=1
7E33  14                0118            INC     D               ;
7E34  00                0119            NOP
7E35  D3FF              0120            OUT     0FFH,A          ;PULS EIN
7E37  1F                0121            RRA
7E38  A7                0122            AND     A
7E39  D0                0123            RET NC
                        0124  ;                                 ----------------------
                        0125
7E3A  3E7F              0126  PULSAUS   LD      A,7FH           ;SCHALTET PULS AUS UND
7E3C  DBFE              0127            IN      A,0FEH          ;KONTOLLIERT AUF BREAK
7E3E  1F                0128            RRA
7E3F  3F                0129            CCF
7E40  D0                0130            RET NC                  ;RETURN FALLS KEIN BREAK
7E41  C35B7F            0131            JP      BREAK
                        0132  ;
                        0133  ;-------------------------------------------------
                        0134  ;-----  *****   SVWAIT: SENDEZEIT ENTSPR. BAUDRATE ***
                        0135  ;-------------------------------------------------
7E44  E509E1            0136  SVWAIT    DEFB    0E5H,09H,0E1H   ;WARTE 32 TAKTE
7E47  3ACD7F            0137            LD      A,(BDRATE)
7E4A  17                0138            RLA
7E4B  D8                0139            RET C                   ;RET FALLS 7200 BAUD
                        0140  ;
                        0141  ;-----  *****   WARTEZEIT FÜR 3200 BAUD  ***** ------
                        0142  ;
7E4C  3E0D              0143            LD      A,0DH           ;230 TAKTE WARTEZEIT
7E4E  3D                0144  WTCONT    DEC     A
7E4F  20FD              0145            JR NZ,  WTCONT
7E51  C0C0              0146            DEFB    0C0H,0C0H
7E53  C9                0147            RET
                        0148
                        0149  ;
                        0150  ;-------------------------------------------------
                        0151  ;-----   *****   LDSUPER: SUPERTAPE LADEROUTINE ****---
                        0152  ;-------------------------------------------------
7E54  21FF7F            0153  LDSUPER   LD      HL,LDEND        ;DER LADEPUFFER WIRD
7E57  0619              0154            LD      B,19H           ;GELOSCHT
7E59  AF                0155            XOR     A
7E5A  77                0156  CLCONT    LD      (HL),A
7E5B  2B                0157            DEC     HL
7E5C  10FC              0158            DJNZ    CLCONT
7E5E  23                0159            INC     HL
                        0160  ;
                        0161  ;-----  *****   LADEN DES PARAMETERBLOCKS ***** -----
                        0162  ;
7E5F  011900            0163            LD      BC,19H          ;BLOCKLÄNGE = 25
7E62  3E2A              0164            LD      A,SYNCPAR
7E64  CDAE7E            0165            CALL    LDBLOC
7E67  A7                0166            AND     A
7E68  C0                0167            RET NZ                  ;RETURN FALLS FEHLER
                        0168  ;
                        0169  ;-----  **** VERGLEICH GELADENER NAME MIT GESUCHTEM **
                        0170  ;
7E69  11E77F            0171            LD      DE,LDNAME
7E6C  21CE7F            0172            LD      HL,SVNAME
7E6F  3AF77F            0173            LD      A,(LDFLAGR)     ;SETZT BDRATE
7E72  32CD7F            0174            LD      (BDRATE),A      ;       "
7E75  4F                0175            LD      C,A
7E76  0610              0176  CPNAME    LD      B,10H           ;NAMENSLÄNGE 16
7E78  7E                0177  CPCONT    LD      A,(HL)          ;
7E79  FE17              0178            CP      17H             ;FALLS "*" SKIP REST
7E7B  280F              0179            JR Z,   SKNAME
7E7D  FE0F              0180            CP      0FH             ;FALLS "?" SKIP CHAR
7E7F  280A              0181            JR Z,   SKCHAR
7E81  1A                0182            LD      A,(DE)
7E82  CB71              0183            BIT     6,C
7E84  CC837F            0184            CALL Z, ASCZX
7E87  BE                0185            CP      (HL)
7E88  3E27              0186            LD      A,27H           ;FALLS FALSCHER NAME
7E8A  C0                0187            RET NZ                  ;RET MIT "B"
7E8B  23                0188  SKCHAR    INC     HL
7E8C  13                0189  SKNAME    INC     DE
7E8D  3E04              0190            LD      A,4H            ;AB B=4 WIRD MIT
7E8F  B8                0191            CP      B               ;SVTYPE VERGLICHEN
7E90  2003              0192            JR NZ,  SKCONT
7E92  21DB7F            0193            LD      HL,SVTYPE
7E95  10E1              0194  SKCONT    DJNZ    CPCONT
                        0195  ;
                        0196  ;-----  *****   PARAM. FÜR DATENBLOCK WERDEN ****-----
                        0197  ;-----  *****   GESETZT UND LADEN/VERIFY AUSGEF. **---
                        0198  ;
7E97  2AF87F            0199            LD      HL,(LDSTART)
```

```
7E9A  ED4BFA7F   0200          LD      BC,(LDLEN)
7E9E  3ACC7F     0201          LD      A,(ZFLAGR)
7EA1  CB57       0202          BIT     2,A             ;NEUE STARTADR?
7EA3  2803       0203          JR Z,   SKADR
7EA5  2A0F7F     0204          LD      HL,(SVSTART)
7EA8  CB4F       0205  SKADR    BIT     1,A             ;LADEN ODER VERIFY?
7EAA  3EC5       0206          LD      A,SYNCDAT
7EAC  2012       0207          JR NZ,  LDVER
                 0208  ;
                 0209  ;-------------------------------------------------
                 0210  ;-----  ***** LDBLOC: LADT DATEN / PARAMETER-BLOCK **
                 0211  ;-------------------------------------------------
7EAE  CDF07E     0212  LDBLOC   CALL    LDSYNC          ;SYNCHRONISIERT
7EB1  CD2A7F     0213  LDNXT    CALL    LDBYTE          ;LADT BYTE
7EB4  77         0214          LD      (HL),A
7EB5  D0         0215          RET     NC              ;WARTE 5 TAKTE
7EB6  EDA1       0216          CPI                     ;INKR LADEREG,DEKR
7EB8  E2D67E   P 0217          JP PO,  LDTEST          ;ZAHLREG:0? SPRUNG
7EBB  A7DBA7     0218          DEFB    0A7H,0DBH,0A7H  ;WARTE 13 TAKTE
7EBE  18F1       0219          JR      LDNXT
                 0220  ;
                 0221  ;-------------------------------------------------
                 0222  ;-----  *****  LDVER: VERIFY-ROUTINE  **** ---------
                 0223  ;-------------------------------------------------
7EC0  CDF07E     0224  LDVER   CALL    LDSYNC          ;SYNCHRONISIERT
7EC3  CD2A7F     0225  VERNEXT CALL    LDBYTE          ;LADT BYTE
7EC6  BE         0226          CP      (HL)            ;VERGLEICHT
7EC7  200A       0227          JR NZ,  ERRVER          ;FEHLER FALLS #
7EC9  0B         0228          DEC     BC              ;DEKR ZAHLREG
7ECA  78         0229          LD      A,B             ;TEST AUF
7ECB  B1         0230          OR      C               ;NULL
7ECC  CAD67E   R 0231          JP Z,   LDTEST          ;0? DANN PRÜFROUT.
7ECF  23         0232          INC     HL
7ED0  A6         0233          DEFB    0A6H            ;WARTE 7 TAKTE
7ED1  18F0       0234          JR      VERNEXT
                 0235  ;
                 0236  ;-----  ***** ERRVER: FEHLER BEI VERIFY ***** -------
                 0237  ;
7ED3  3E12       0238  ERRVER   LD      A,12H           ;RET MIT ")"
7ED5  C9         0239          RET
                 0240  ;
                 0241  ;-------------------------------------------------
                 0242  ;***** LDTEST ***** PRÜFSUMMENTEST
                 0243  ;-------------------------------------------------
7ED6  FDE5       0244  LDTEST   PUSH    IY              ;LADE BC MIT ERRECHN.
7ED8  C1         0245          POP     BC              ;PRÜFSUMME
7ED9  CD2A7F     0246          CALL    LDBYTE          ;LADE HL MIT
7EDC  6F         0247          LD      L,A             ;GELESENER PRÜFSUMME
7EDD  E3E3       0248          DEFB    0E3H,0E3H       ;WARTE 59 TAKTE
7EDF  E5E1       0249          DEFB    0E5H,0E1H       ;       "
7EE1  CD2A7F     0250          CALL    LDBYTE
7EE4  67         0251          LD      H,A
7EE5  FD210040   0252          LD      IY,4000H
7EE9  AF         0253          XOR     A
7EEA  ED42       0254          SBC     HL,BC
7EEC  C8         0255          RET Z                   ;RET MIT " " FALLS OK
7EED  3E2B       0256          LD      A,2BH           ;RET MIT "F" FALLS
7EEF  C9         0257          RET                     ;FEHLER
                 0258  ;
                 0259  ;-------------------------------------------------
                 0260  ;***** LDSYNC ***** SYNCHRONISIERUNGSROUTINE
                 0261  ;-------------------------------------------------
7EF0  1E00       0262  LDSYNC   LD      E,00H           ;INIT LADEBYTE
7EF2  C5         0263          PUSH    BC              ;RETTE BC
7EF3  4F         0264          LD      C,A             ;C= ABSCHLUßMARKE
                 0265  ;
                 0266  ;------*****  WARTE AUF EIN SYNCHAR **** ----------
                 0267  ;
7EF4  A7A6       0268  LDNXBIT DEFB    0A7H,0A6H       ;WARTE 11 TAKTE
7EF6  E509E1     0269  SYNEU   DEFB    0E5H,9H,0E1H    ;WARTE 32 TAKTE
7EF9  CBC3       0270          SET     0,E             ;LADE EIN BIT
7EFB  CD2C7F     0271          CALL    LDBIT           ;       "
7EFE  FE16       0272          CP      SYNCHAR         ;FALLS NOCH KEIN
7F00  20F2       0273          JR NZ,  LDNXBIT         ;SYNCHAR:NACHSTES BIT
                 0274  ;
                 0275  ;-------*****  TEST AUF WEITERE SYNCHAR **** -------
                 0276  ;
7F02  A7A7D8     0277          DEFB    0A7H,0A7H,0D8H  ;WARTE 13 TAKTE
7F05  060A       0278          LD      B,0AH           ;ZÄHLER AUF 10
7F07  3435A6     0279  SYCONT  DEFB    34H,35H,0A6H    ;WARTE 29 TAKTE
7F0A  CD2A7F     0280          CALL    LDBYTE          ;LADE BYTE
7F0D  FE16       0281          CP      SYNCHAR         ;KEIN SYNCHAR? DANN
7F0F  20E3       0282          JR NZ,  LDNXBIT         ;SUCHE WEITER
7F11  10F4       0283          DJNZ    SYCONT          ;10*SYNCHAR ERKANNT?
                 0284  ;
                 0285  ;-------*****  ÜBERLESEN FOLGENDER SYNCHARS **** ---
                 0286
7F13  1802       0287          JR      SYEIN           ; AUS ZEITGRÜNDEN
7F15  3435       0288  SYIGN   DEFB    34H,35H         ;WARTE 22 TAKTE
7F17  3435       0289  SYEIN   DEFB    34H,35H         ;WARTE 22 TAKTE
7F19  CD2A7F     0290          CALL    LDBYTE          ;LADE BYTE
7F1C  FE16       0291          CP      SYNCHAR         ;WEITER FALLS KEIN
7F1E  28F5       0292          JR Z,   SYIGN           ;SYNCHAR
                 0293  ;
                 0294  ;-------*****  TEST AUF ABSCHLUSSMARKE **** ----------
                 0295  ;
7F20  B9         0296          CP      C               ;SUCHE WEITER FALLS
7F21  20D3       0297          JR NZ,  SYNEU           ;FALSCHE MARKE
7F23  FD210000   0298          LD      IY,0000H        ;INIT PRÜFSUMME
7F27  C1         0299          POP     BC
7F28  AF         0300          XOR     A
7F29  C9         0301          RET
                 0302  ;
                 0303  ;-------------------------------------------------
                 0304  ;**** LDBYTE: LADT EIN BYTE  BZW. BIT VOM BAND ****
                 0305  ;-------------------------------------------------
7F2A  1E80       0306  LDBYTE   LD      E,80H           ;SETZT FLAG FÜR 8 BIT
7F2C  CD617F     0307  LDBIT    CALL    LDWAIT          ;WARTET ENTSP. BDRATE
7F2F  3E7F       0308          LD      A,7FH           ;LIEST EINGANGSSIGNAL
7F31  DBFE       0309          IN      A,(0FEH)        ;UND VERGLEICHT MIT
7F33  BA         0310          CP      D               ;DEM VORAUSGEGANGENEN
7F34  F5         0311          PUSH    AF
7F35  57         0312          LD      D,A             ;SPEICHERT NEUES SIG-
7F36  3E7F       0313  FLWAIT   LD      A,7FH           ;NAL UND WARTET AUF
7F38  DBFE       0314          IN      A,(0FEH)        ;FLANKE
7F3A  BA         0315          CP      D
7F3B  28F9       0316          JR Z,   FLWAIT
7F3D  57         0317          LD      D,A             ;TEST AUF BREAK
7F3E  1F         0318          RRA
7F3F  301A       0319          JR NC,  BREAK
                 0320  ;
                 0321  ;-----  *****  AUSWERTUNG DES GELESENEN SIGNALS ****-
                 0322  ;
7F41  F1         0323          POP     AF
7F42  2806       0324          JR Z,   LDEINS          ;ALTES = NEUES? EINS
                 0325  ;
                 0326  ;-----  *****   NULL ERKANNT  **** -----------
                 0327  ;
7F44  C8C800     0328  LDNULL   DEFB    0C8H,0C8H,0H    ;WARTE 14 TAKTE
7F47  AF         0329          XOR     A               ;LÖSCHE CARRY
7F48  1805       0330          JR      LDCONT
                 0331  ;
                 0332  ;-----  *****   EINS ERKANNT  **** -----------
                 0333  ;
7F4A  37         0334  LDEINS   SCF                     ;SETZE CARRY
7F4B  D3FF       0335          OUT     0FFH,A          ;DISPLAYAUSGABE
7F4D  FD23       0336          INC     IY              ;INKR. PRÜFS.-ZÄHLER
7F4F  CB18       0337  LDCONT   RR      E               ;SPEICHER CARRY IN E
7F51  7B         0338          LD      A,E
7F52  D8         0339          RET C                   ;RET BEI ENDFLAG
7F53  E3E3       0340          DEFB    0E3H,0E3H       ;WARTE 81 TAKTE
7F55  E5E1       0341          DEFB    0E5H,0E1H
7F57  3435       0342          DEFB    34H,35H
7F59  18D1       0343          JR      LDBIT           ;LADE NACHSTES BIT
                 0344  ;
                 0345  ;-------------------------------------------------
                 0346  ;-----  *****   BREAKROUTINE    ***** ----------
                 0347  ;-------------------------------------------------
7F5B  FD210040   0348  BREAK    LD      IY,4000H        ;SETZE ZX81-INDEX
7F5F  CF         0349          RST     08H             ;FEHLERROUTINE MIT
7F60  0C         0350          DEFB    0CH             ;CODE: "D"
                 0351  ;
                 0352  ;-------------------------------------------------
                 0353  ;-----  *****   LDWAIT: PRÜFZEIT ENTSPR. BAUDRATE ***
                 0354  ;-------------------------------------------------
7F61  3E02       0355  LDWAIT   LD      A,02H           ;WARTESCHLEIFE FÜR
7F63  3D         0356  WT7200   DEC     A               ;7200 BAUD
7F64  20FD       0357          JR NZ,  WT7200          ;INSGES. 62 TAKTE
7F66  3ACD7F     0358          LD      A,(BDRATE)
7F69  17         0359          RLA
7F6A  D8         0360          RET C
                 0361  ;                               ----------
7F6B  3E11       0362          LD A,011H               ;ZUSATZL. WARTEZEIT
7F6D  3D         0363  WT3600   DEC     A               ;FÜR 3600Bd: 306 T
7F6E  20FD       0364          JR NZ,  WT3600          ;INSGES.: 362 TAKTE
7F70  3435       0365          DEFB    34H,35H
7F72  C9         0366          RET
                 0367  ;
                 0368  ;-------------------------------------------------
                 0369  ;-----  *****   ZX81-CODE / ASCII UMFORMUNG ***** ---
                 0370  ;-------------------------------------------------
7F73  E63F       0371  ZXASC    AND     3FH             ;MASKIERT ZEICHEN
7F75  C61B       0372          ADD     1BH             ;ADDIERT 27 UND RET
7F77  FE41       0373          CP      41H             ;FALLS BUCHSTABE
7F79  D0         0374          RET NC
7F7A  E5         0375          PUSH    HL              ;RETTET HL
7F7B  267F       0376          LD      H,7FH           ;HI-BYTE DER TABADR
7F7D  C688       0377          ADD     A,88H           ;BERECHN. LO-BYTE
7F7F  6F         0378          LD      L,A
7F80  7E         0379          LD      A,(HL)          ;A=ASCII-ZEICHEN
7F81  E1         0380          POP     HL
7F82  C9         0381          RET
                 0382  ;
                 0383  ;-------------------------------------------------
                 0384  ;-----  *****   ASCII / ZX81-CODE UMFORMUNG ***** ---
                 0385  ;-------------------------------------------------
7F83  FE60       0386  ASCZX    CP      60H             ;BILDET DIE KLEINEN
7F85  3802       0387          JR C,   GROSS           ;AUF DIE GROSSEN
7F87  D620       0388  KLEIN    SUB     020H            ;BUCHSTABEN AB
7F89  D620       0389  GROSS    SUB     020H
7F8B  E63F       0390          AND     3FH
7F8D  D621       0391          SUB     21H
7F8F  FE1A       0392          CP      1AH
7F91  3003       0393          JR NC,  SONDER          ;RET FALLS BUCHSTABE
7F93  C626       0394          ADD     26H             ;SONST SUCHE SONDER-
7F95  C9         0395          RET                     ;ZEICHEN IN TABELLE
7F96  C641       0396  SONDER   ADD     41H
7F98  E5         0397          PUSH    HL              ;RETTE HL UND BC
7F99  C5         0398          PUSH    BC
7F9A  21CB7F     0399          LD      HL,TABEND       ;ENDADR. DER TABELLE
7F9D  012600     0400          LD      BC,26H          ;LÄNGE DER TAB: 38
7FA0  EDB9       0401          CPDR                    ;DURCHSUCHE TABELLE
7FA2  79         0402          LD      A,C             ;A= ZX81-CODE
7FA3  C1         0403          POP     BC
7FA4  E1         0404          POP     HL
7FA5  C9         0405          RET
                 0406  ;
                 0407  ;-------------------------------------------------
                 0408  ;------ *****   ASCII-UMFORMTABELLE  ***** ---------
                 0409  ;-------------------------------------------------
                 0410
      (7FA6)     0411          ORG     7FA6H
7FA6  20585C5D   0412  ASCTAB   DEFB    20H,58H,5CH,5DH
7FAA  5E5F2123   0413          DEFB    5EH,5FH,21H,23H
7FAE  25274022   0414          DEFB    25H,27H,40H,22H
7FB2  26243A3F   0415          DEFB    26H,24H,3AH,3FH
7FB6  28293E3C   0416          DEFB    28H,29H,3EH,3CH
7FBA  3D2B2D2A   0417          DEFB    3DH,2BH,2DH,2AH
7FBE  2F3B2C2E   0418          DEFB    2FH,3BH,2CH,2EH
7FC2  30313233   0419          DEFB    30H,31H,32H,33H
7FC6  34353637   0420          DEFB    34H,35H,36H,37H
7FCA  3839       0421          DEFB    38H,39H
                 0422
                 0423
7FCC  (0000)     0424          END
```

Warte mal ...

Wenn Sie den Z80 X Takte warten lassen wollen, und X liegt zwischen 9 und 4113, dann können Sie die Warteroutine nach folgendem Schema bestimmen:

```
      LD A.N        3E ...
LOOP  DEC A         3D
      JR NZ LOOP    20 FD
      (DUMMY)       .. .. ..
```

N und die Dummy-Befehle müssen dann bestimmt werden. Aus

(X—2)/16 = Z Rest R

ergibt sich der ganzzahlige Anteil Z und der Rest R.

R	Dummies	N
0	—	Z
1	A6D8D8	Z—1
2	A6A6A7	Z—1
3	A6A6D8	Z—1
4	A7	Z
5	C0	Z
6	3435	Z—1
7	A6	Z
8	A7A7	Z
9	A7D8	Z
10	C0C0	Z
11	A6A7	Z
12	A6D8	Z
13	A7A7D8	Z
14	A6A6	Z
15	A6A7A7	Z

Die Dummies und die Schleife sind so ausgelegt, daß nur das AF-Register beeinflußt wird.

Langsam, aber gründlich

Casio FP 200: Hand-Held mit Statistik-Funktionen

Peter Hagemann

Der Anwenderkreis von Hand-Held-Computern wächst ständig. Nun bietet Casio mit dem FP 200 für unter 1000 D-Mark ein wohl speziell für den Einsteiger interessantes Gerät an. Kann dieser Computer neben seinen deutlich teureren Konkurrenten (EPSON HX-20, TRS 80 Modell 100) bestehen? Der c't-Test offenbart die Besonderheiten des FP 200.

In der Grundausstattung bekommt man den FP 200 ohne Koffer, ohne Tasche und ohne Netzteil. Zum Lieferumfang gehört allerdings ein Batteriesatz, eine Bedienungsanleitung, ein Bezugshandbuch und eine Kommando-Liste. Die in deutsch gehaltene Literatur erklärt auch dem ungeübten Anwender, wie er seinen FP 200 zum Leben erweckt.

Der Computer ist mit seinen Maßen von 310 x 220 x 56 mm in allen drei Dimensionen geringfügig größer als der HX-20; das Format 'DIN A 4' trifft also nicht mehr zu. Trotzdem behält das stabile Gehäuse 'Hand-Held-Format'.

Als erstes fällt am FP 200 auf, daß das asymmetrisch eingebaute LCD eigentlich auch doppelt so groß sein könnte. Die Anzeige hätte dann mit 40 x 8 Zeichen der des TRS 80 Modell 100 entsprochen. Aber auch so ist es erfreulich, daß mit 20 Spalten in 8 Zeilen doppelt soviele Zeichen dargestellt werden können wie auf dem LCD des HX-20.

Jedes Zeichen stellt der FP 200 in einem 8 x 8-Feld dar. Die dazu benötigten 160 x 40-Punkte lassen sich darüber hinaus auch einzeln setzen, löschen oder prüfen.

Wie bei seinen Mitbewerbern ist beim FP 200 ein Steller vorhanden, mit dem sich der Kontrast des LCDs, entsprechend dem Betrachtungswinkel, anpassen läßt.

Tastatur

Hier wurde am verkehrten Ende gespart. Die mechanisch sehr einfach geratene Konstruktion besitzt weder 'Auto-Repeat' noch eine 'Roll-Over'-Funktion; das heißt bevor ein neuer Tastendruck angenommen wird, muß die zuletzt gedrückte Taste losgelassen werden. Selbst für 'mittelschnelle' Schreiber ist dieses System zu langsam.

Zur Verfügung stehen der ASCII-Zeichensatz und einige Grafikzeichen. Die deutschen Umlaute und das 'ß' sind nicht darstellbar. Ansonsten werden Sonderfunktionstasten durch Größe oder Farbe vom eigentlichen Tastenfeld abgehoben.

Fünf Funktionstasten sind doppelt (SHIFT) mit bis zu je 15 Zeichen frei belegbar. Wem das durch Umschaltung erreichbare, integrierte Zehnerfeld nicht ausreicht, der kann eine zusätzliche, externe Zehnertastatur anschließen (Option).

Schnittstellen

Ein Blick auf die Rückseite des Gerätes läßt folgende Schnittstellen erkennen:

Kassettenrecorder-Interface (CMT)

Über die 8polige Buchse läßt sich ein handelsüblicher Rekorder anschließen. (Der Kauf eines Spezialgerätes ist also nicht erforderlich.)

RS-232 C-Interface

Der Anschluß erfolgt ebenfalls über eine 8polige Buchse. Das ist zwar nicht unbedingt normgerecht, spart aber (wie beim HX-20) Steckerkosten.

Selbst CASIO befürchtet (laut Handbuch) Verwechslungen der CMT und der RS-232 C-Buchse. Es ist daher unverständlich, warum für den Rekorder keine andere Buchse vorgesehen wurde. Konstruktiv müßte das leicht möglich sein, da beim CMT trotz REMOTE nur fünf Stifte belegt sind.

CENTRONICS-kompatible Parallelschnittstelle

Hierzu läßt sich nicht viel sagen, außer, daß die Haltebügel für den Stecker störend vom Gerät abstehen (nicht einklappbar).

Anschluß für ein CASIO-Floppy-Disk-Laufwerk

Diese Buchse befindet sich hinter dem Batteriekasten, der vor dem Anschließen der Floppy entfernt werden muß.

Für diejenigen, die sich über das RS-232 C-Interface gefreut haben, hier nun der Wermutstropfen: Sowohl die CMT- als auch die RS-232 C-Schnittstelle arbeiten mit einer festgelegten Übertragungsgeschwindigkeit von 300 Baud. Eine Umstellung ist nicht möglich. Auch eine so komfortable Einstellung der Übertragungsdaten wie beim HX-20 oder beim TRS 80 Model 100 ist beim FP 200 nicht durchführbar.

Alle Interfaces, bis auf die Recorder-Schnittstelle, arbeiten nur bei angeschlossenem Netzgerät. Hierdurch wird die Mobilität des Systems doch etwas eingeschränkt.

In der Grundausstattung wird der FP 200 zwar mit 32K ROM, aber nur mit 8K RAM geliefert. Wem das zu wenig ist, der kann den Speicher in 8K-Schritten aufrüsten. CASIO bietet hierzu Module an, die ohne Mithilfe eines Fachhändlers vom Anwender eingesetzt werden können. Geplant ist auch eine zur Zeit nicht lieferbare 8K-ROM-Erweiterung (nach gleichem System).

Wer einen zünftigen Steckplatz für die Speicher-Module erwartet hat, der wird enttäuscht. Die Module werden über Leitgummi-Kontaktleisten auf die vergoldeten Leiterbahnen gedrückt und dort arretiert (Bild 1).

Die Hauptstromversorgung des Gerätes erfolgt entweder durch das Netzteil (Option) oder durch vier in den Batteriekasten eingelegte Mignon-Batterien. Im letzteren Fall beträgt die Betriebsdauer laut Herstellerangaben bei Alkali-Batterien und 32K RAM-Ausbau nur circa elf Stunden. Diese Tatsache und die Betriebseinschränkung der Schnittstellen machen das Netzgerät zu einer dringend erforderlichen Anschaffung.

Etwas riskant erscheint die Art der Speicherpufferung, da CASIO eine Methode ohne NiCd-Akku gewählt hat. Getrennt vom Haupt-Batteriekasten existiert eine weitere Batteriehalterung für zwei 'Speicherschutz-Batterien'.

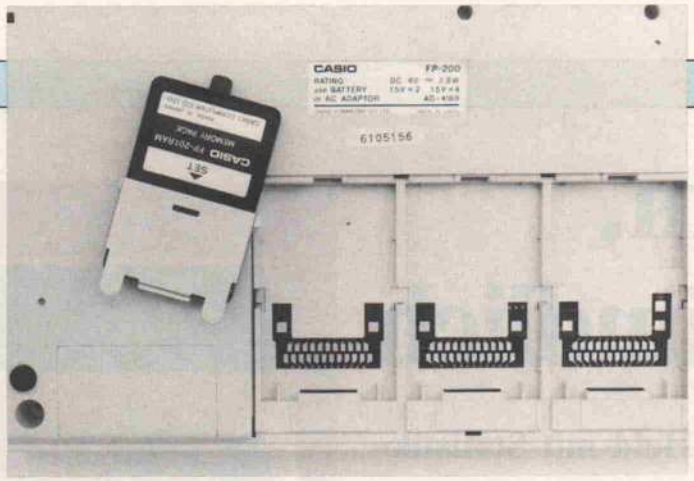
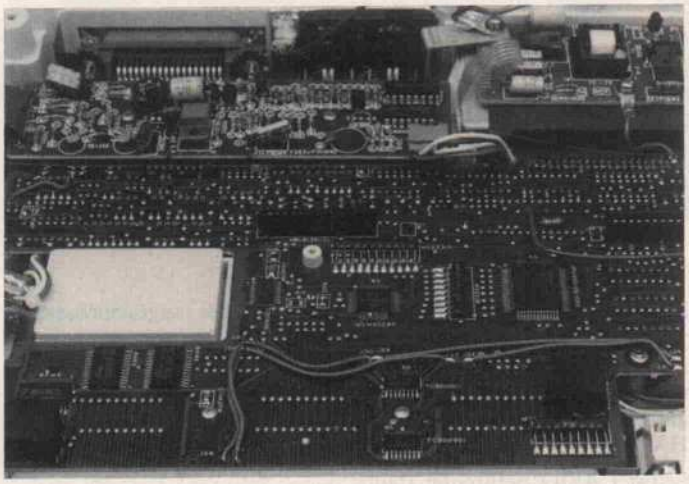


Bild 1. Eine eigenwillige Lösung für Speichererweiterungen: Die Kontaktierung erfolgt beim Einsetzen der Module über Leitgummi, das auf vergoldete Leiterbahnen drückt.

Bild 2. Bemerkenswert an der Platine: Enge Leiterbahnführung und Flat-Pack-ICs.



Hierdurch wird der RAM-Inhalt auch bei abgeschaltetem FP 200 erhalten.

Leider gibt es keine Anzeige und kein Meßgerät, das etwas über den Zustand der Schutz-Batterien aussagt. Man muß also gelegentlich (hoffentlich rechtzeitig) Messungen anstellen oder einen Austausch auf Verdacht durchführen. CASIO spricht von einer 6monatigen Lebensdauer der Batterien.

Ein Wechsel der Speicherschutz-Batterien zerstört den RAM-Inhalt. Nach dem Wechsel beziehungsweise bei der ersten Inbetriebnahme muß das RAM durch Eingabe von 'RESET' <RETURN> initialisiert werden. Der FP 200 'merkt' hierbei gleichzeitig, wieviel Speicher-Module in seinem Bauch eingebaut sind.

Die 'Datenverarbeitung' wird beim FP 200 von einer 80C85-CPU durchgeführt. Die Quarzfrequenz beträgt 6,144 MHz, was einem Systemtakt von rund 3 MHz entspricht.

CETL ...

Im Gegensatz zu anderen Rechnern kann man beim FP 200 durch einen 'MODE'-Schalter festlegen, ob man mit BASIC oder mit CETL arbeiten will. Bei CETL handelt es sich um ein einfaches Tabellenkalkulationsprogramm. Dadurch wird in gewissem Umfang eine Geräteanwendung ohne Programmierkenntnisse ermöglicht.

CETL läßt sich durch 16 leicht erlernbare Befehle handhaben. Die Eingaben erfolgen im Dialog mit dem Rechner (Tabelle 1).

<A>uto	<B>lank	<C>omp
<D>elete	<F>ind	<G>et
<I>nsert	<J>ump	<K>ill
<L>ist	<M>ove	<N>ew
<P>ut	<R>ename	<S>ort
<T>able		

Tabelle 1. Befehlssatz von CETL.

Der Umfang numerischer oder alphanumerischer Tabellen hängt nur von der jeweiligen Speicherkapazität ab. Die Berechnung der maximal möglichen Datensätze ist in der Bedienungsanleitung verständlich beschrieben.

CETL verwaltet im RAM bis zu zehn verschiedene Dateien. Als Ergebnis bekommt man einen formatierten Ausdruck. Jede einzelne Dateiinformation läßt sich aber auch auf dem LCD betrachten. Korrekturen und Eingaben sind genauso leicht möglich wie Verknüpfungen der Datenfelder durch die Grundrechenarten. CETL kann komplizierte Bearbeitungsaufträge an BASIC übergeben. Die geänderten Daten können anschließend wieder von CETL genutzt werden.

... und BASIC

Wer einen FP 200 in der Grundversion gekauft hat und BASIC startet, wird mit der Meldung: '1902 Bytes Free' überrascht. Wo ist der Rest von den 8K RAM geblieben?

Circa 2,3K werden unwiderruflich an das Betriebssystem abgetreten. Die restlichen 5,7K teilt der FP 200 automatisch zwischen BASIC, Stringvariablenbereich und CETL auf.

Mit 'CLEAR' und 'AREA' lassen sich Bereichsänderungen vornehmen. Trotzdem erhält man nie mehr als etwa 5,7K für den Betrieb mit BASIC. Neben dem Netzteil zeichnet sich hier also eine Speichererweiterung als unbedingt erforderlich ab.

Das BASIC selbst weist einige Besonderheiten auf. So sind zum Beispiel Variablennamen mit 255 signifikanten Stellen möglich. Wer jetzt noch Schwierigkeiten bei der Namensgebung hat ist selber Schuld.

Felder lassen sich aber leider nur dreidimensional festlegen. Der FP 200 ist sehr genau! Bei Variablen einfacher Genauigkeit werden intern neun Stellen verwendet, von denen sechs angezeigt werden können. Bei doppelter Genauigkeit werden intern sogar 19 Stellen verarbeitet. Der Speicherplatzbedarf steigt dadurch aber auf sechs beziehungsweise elf Bytes pro Wert.

Bei so viel Genauigkeit gibt sich der FP 200 mit einfachen 'Integers' gar nicht erst ab. Eine Integerdefinition ist ebenso wie die im Handbuch beschriebene 'erweiterte' Genauigkeit nicht möglich.

Der hohe Speicherplatzbedarf der Variablen und die fehlenden Integer-Verarbeitung macht sich natürlich deutlich im Benchmark-Test bemerkbar (Tabelle 2).

Obwohl die Benchmark-Tests nicht das 'A & O' der Computer-Bewertung darstellen, werden extreme Werte des FP 200 vermutlich manchen Kaufinteressenten abschrecken.

Daß der FP 200 gut mit Zahlen umgehen kann, steht außer Frage. Als einer der wenigen Rechner bietet er sogar implementierte Statistik-Befehle (Tabelle 3). Dadurch wird der Programmieraufwand für diese Problemstellung erheblich reduziert.

Überhaupt ist der Befehlssatz recht umfangreich. Alle Befeh-

Rechner	BMT1	BMT2	BMT3	BMT4	BMT5	BMT6	BMT7	BMT8
alphaTronic PC	2,2	5,3	15,4	16,7	18,1	31,0	42,6	17,8
Apple II Plus	1,4	8,4	15,8	17,6	19,0	28,4	45,0	10,4
TRS 80 M 1 L2	2,8	11,2	27,0	27,8	31,0	50,6	78,0	11,8
TRS 80 M 100	3,7	9,8	26,6	29,7	31,4	46,8	62,8	30,9
CASIO FP 200	4,6	17,6	55,4	56,4	60,7	95,5	137,0	36,9

Tabelle 2. Benchmark-Test (Zeitangabe in Sekunden)

Funktion	Bedeutung
CNT	Anzahl der Eingabepunkte für statistische Verarbeitung
SUMX	Summe der X-Daten
SUMY	Summe der Y-Daten
SUMX 2	Summe der Quadraten der X-Daten
SUMY 2	Summe der Quadraten der Y-Daten
SUMXY	Summe der Produkte der X- und Y-Daten
MEANX	Mittelwert der X-Daten
MEANY	Mittelwert der Y-Daten
SDX	Probenstandardabweichung der X-Daten
SDY	Probenstandardabweichung der Y-Daten
SDXN	Standardabweichung von der Gesamtheit der X-Daten
SDYN	Standardabweichung von der Gesamtheit der Y-Daten
LRA	Koeffizient für lineare Regression
LRB	Konstantenausdruck für lineare Regression

Tabelle 3. Statistik-Funktionen des FP 200

	HX-20	Modell 100	FP 200
Stromversorgung	Akku	Batt/Netz	Batt/Netz
Datensicherung	Akku	Batt/Akku	Batterie
LCD	20 x 4	40 x 8	20 x 8
Tastatur	deutsch	ASCII	ASCII
Programmspeicherung	5	19	10 + 10 Datei
Programme BASIC +	Monitor	Text, Adressen Termin, TELCOM	CETL
Dokumentation	deutsch	englisch	deutsch
Prozessor	2 x 6301	80C85	80C85
Editieren	Bildschirm	extra Modus	extra Modus
Variablenidentifikation	16	2	255
Variablendef. String +	INT, SNG, DBL	INT, SNG, DBL	SNG, DBL
RS-232C-Interface	einstellbar	einstellbar	fest
CENTRONICS	nein	ja	ja

Tabelle 4. Der FP 200 im Vergleich

le für die Vorbereitung von RANDOM-Dateien bei Floppy-Betrieb (FIELD, LSET, RSET, usw.) sind ebenso wie RENUM, OPTION BASE und DEFFN bereits im ROM enthalten. Es verwundert aber, daß eine so gebräuchliche Funktion wie INSTR nicht vorgesehen ist.

BASIC verwaltet, ähnlich wie CETL, zehn verschiedene Programmbereiche. Allen Programmbereichen stehen sämtliche Variablenwerte zur Verfügung, denn ein 'PROG-Wechsel' oder ein 'RUN' zerstört die Variableninhalte nicht. Auch nach dem Aus- und Einschalten bleiben alle Werte erhalten. Dadurch lassen sich Funktionen leicht auf verschiedene Programme aufteilen. Man muß nur aufpassen, daß der Variablenspeicherbedarf nicht zu groß wird.

Alle Programmebenen können gemeinsam durch ein Paßwort geschützt werden. Ein LIST, LOAD oder SAVE wird dann unmöglich.

Betrüblich ist es, daß man mit den Cursor-Tasten auf dem LCD zwar beliebig 'hin- und herfahren' kann, aber trotzdem eine Korrektur nicht möglich ist. Um zu korrigieren, muß der 'EDIT-Modus' aufgerufen werden.

Wer bei der Taste 'INSERT' an ein komfortables Einfügen denkt, wird ebenfalls enttäuscht. Es können lediglich Leerzeichen eingefügt werden, die man in einem weiteren Arbeitsgang überschreiben kann.

Fazit

Die gewonnenen Erkenntnisse führen zu folgendem Ergebnis: Für unter 1 000,— DM erhält man relativ 'viel' Computer.

Durch CETL wird auch dem Anfänger eine sofortige Anwendungsmöglichkeit geboten. Die sicherlich verbesserungsfähige Hardware sollte unter Berücksichtigung des Preises mit der Konkurrenz verglichen werden.

Die Aufstellung (Tabelle 4) soll dabei helfen. Zeitkritische Anwendungen dürften mit dem CASIO nicht zu lösen sein. Die Tastatur ist für eine schnelle Eingabe nicht geeignet. □

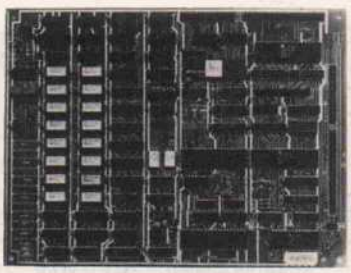
Ergebnisse auf einen Blick:

- ⊕ günstiger Preis
- ⊕ integriertes Tabellenkalkulationsprogramm
- ⊕ CENTRONICS und RS-232 C
- ⊕ Speicher kann ohne Werkstatt erweitert werden
- ⊕ integrierte Statistik-Befehle
- ⊕ lange Variablennamen
- ⊖ zu einfache ASCII-Tastatur
- ⊖ nicht überwachter Speicherschutz
- ⊖ langsame Ausführungszeit
- ⊖ nur 300 Baud bei CMT und RS-232 C
- ⊖ Grundversion nur 5,7K freies RAM
- ⊖ Schnittstellen nur mit Netzteil anzusprechen (außer CMT)
- ⊖ kein Bildschirmeditor

BULLET — das CP/M-Geschoß

Einer der schnellsten und leistungsfähigsten CP/M-Rechner für CP/M 3,0 (Plus) und 2,2

System-Lieferung möglich:
mit 2 mal 5,25-Zoll-Laufwerken je 1 MB oder 2 mal 5,25-Zoll-Laufwerken je 1,6 MB oder 5,25-Zoll-Laufwerk und eingeb. Harddisk 10 MB oder 2 mal 8-Zoll-Laufwerke DS/DD je 1,6 MB oder 8-Zoll-Laufwerk plus Harddisk ab 10 MB



20 x 27 cm

- CP/M-3,0 (Plus)-Software ist im Lieferumfang enthalten
- CP/M 2,2 mit 47 KB-RAM-Pseudo-Floppy möglich
- CP/M 3,0 mit 60 KB freiem Benutzerspeicher
- Z80 A — 4 MHz
- Durch DMA und Disk-Track-Buffering superschnell bei Disk-I/O: 64 KB werden in 14 sec mit Verify auf Disk geschrieben und in 10 sec gelesen, MBASIC in 3,3 sec gelesen (siehe Vergleichstest in BYTE Jan./Nov. 82)
- Anschluß für Parallel- und Seriell-Printer, Terminal Anschlußmöglichkeit für ECB-Bus-Karten, weitere V24, RAM-Floppy etc.

Universell einsetzbar für:
- OEM's
- Praxis und Hobby
- Entwicklung
- Universitäten und Institute

Fachhändler gesucht

- Verwaltung und Anschluß für vier Floppys 5,25-Zoll (40, 80 oder 77 Spur) und Anschluß für vier Floppys 8-Zoll (SS/SD oder DS/DD) gleichzeitig, Anschluß über Flachbandkabel Harddisk-Anschluß serienmäßig
- Kopieren der Software von 5,25-Zoll auf 8-Zoll und Kopieren der Software von 8-Zoll- auf 5,25-Zoll-Diskette
- alle gängigen Programmsprachen lauffähig
- Lieferung als Platine oder als Komplettsystem — Sie bestimmen, wir liefern —
- Kundenspezifische Applikationen möglich
- Anschluß-Platine für 8086 für MS-DOS und RAM-Floppy bis 1 MB vorgesehen.
- Convertierungsbullet zum Lesen und Kopieren von Daten fast aller Disketten anderer Computer.

Ausführliche Unterlagen und Anwendungsbeispiele bei:

EPS
M. Mandt
Ilmspaner Straße 29, D-6971 Großrinderfeld
Telefon (0 93 49) 2 71-12 71, Telex 6 89 549 EPS D

Detlef Grell

In puncto Kleincomputerpatriotismus sind uns die Briten vermutlich uneinholbar voraus. Hat zumindest unsere Konsumer-Industrie Geräte, die man an Stelle japanischer Produkte kaufen kann, so sind Computer aus Deutschland immer noch rar. Und je länger die deutsche Industrie herumtrödelte, desto schwieriger wurde es natürlich, noch eine 'ökonomische Nische' auszumachen. Mit dem gesamten Konzept, das hinter MAX 1 steckt, scheint tatsächlich eine Marktlücke gefunden zu sein.

MAX aus Deutschland

Vielseitiger Kleincomputer, auch zum Selbstbau

Auf den ersten Blick ist MAX ein Hand-Held-Computer. Komfortable deutsche Schreibmaschinentastatur und zweizeiliges LC-Display drängen einem diesen Eindruck auf. BASIC ist resident vorhanden, ebenso ein Kassetten-Interface und eine Centronics-Schnittstelle, und das RAM ist gepuffert.

Das Format hingegen tendiert deutlich mehr zu DIN A3 als zum obligaten Hand-Held-A4. Und wenn man ihn in der Alu-Blech-Gehäuse-Version in der Hand 'held', ist die Verwirrung komplett: rund drei Kilogramm massive Solidität. (Erinnerungen an deutsche Wertarbeit werden wach.)

Vielseitig

Mit MAX kann man (laut Prospekt) rechnen, schreiben, kreativ spielen, programmieren, wecken, Zeit anzeigen, messen, steuern und regeln. Dem Leser bleibe überlassen, wie er MAX nun bezeichnen möchte.

MAX gibt es als Bausatz und als Fertiggerät. Außerdem kann er durch spezielle Steckmodule einfach erweitert werden. Man bekommt ihn auch im Kunststoffgehäuse und in verschiedenen Ausbaustufen, sowohl was Hardware als auch Software anbelangt.

Und es gibt etwas ganz Neues: Der Kauf eines Bausatzes ist mit einem eintägigen Selbstbau-Seminar gekoppelt. Nach dem Motto 'morgens gebracht, abends gemacht' wird dem Käufer garantiert, ein fertiges und funktionierendes Gerät mit nach Hause zu nehmen.

Alle, für die nur zwei Prozessoren Weltgeltung besitzen, werden den 6803, der den MAX

MAX' Anatomie

beseelt, mit einer gewissen Mißbilligung betrachten. Nun, auch wenn wir durchaus 'anderen' Prozessoren tolerant gegenüberstehen: der Weg zu CP/M und APPLE-Software ist erstmal verbarrikadiert. Allerdings liegen MAX' Hauptanwendungen auch ganz woanders, so daß man ohnehin mehr auf spezielle Software oder Hausgemachtes zurückgreifen wird.

Der eigentliche Rechner befindet sich auf einem Mother-Board, und somit gibt es natürlich auch die geliebten SLOTS, allerdings ganz anders. Erweiterungen werden von außen als Steckmodule in das Gehäuse eingeschoben, und zwar ohne Öffnen des Hauptgerätes. Diese kleinen Baugruppenträger mit Steckerleiste für Wrap-Pfosten haben eine eigene Frontplatte, und — wo erforderlich — eine Canon-Buchse für den Kontakt nach außen.

Da es MAX als Selbstbaugerät gibt, ist ein luftiger Aufbau, wie er hier praktiziert wird, sicher günstiger als eine vollgequetschte Multilayer-Platine. Derzeit sind auf der Platine maximal 16K statisches RAM unterzubringen, drei PIAs (6821), drei EPROMs (2764), ein Uhrenbaustein (der standardmäßig dazugehört) und diverse 'kleine' ICs. Nicht zu vergessen die Puffer-Akkus, die allerdings nicht den Betrieb des Rechners unabhängig vom Netz ermöglichen, sondern lediglich der Daten- und Programmsicherung dienen. Auf der Hauptplatine befindet sich auch die Elektronik für das Kassetten-Interface (über fünfpolige DIN-Buchse herausgeführt) und für die Centronics-Schnittstelle.

Zusätzlich zum Mother-Board ist ein Netzteil vorhanden. An der Rückseite des Gehäuses befindet sich eine Kaltgeräte-Steckdose zum Anschluß eines Netzkabels. Auf der Netzteilplatine ist auch das Relais zum Ein-/Ausschalten (im Schaltuhrbetrieb oder per BASIC) einer in der Rückwand eingebauten Schuko-Netzsteckdose untergebracht (siehe auch Bild 1).

Kennenlernen

MAX kommt dem Einsteiger entgegen. Nach dem Einschalten meldet er sich mit 'Hallo, alles klar, ich bin bereit ...', und in der Zeile darunter findet man auch schon das Hauptmenü mit Abkürzungen, die in ihrer Reihenfolge unmittelbar den acht Funktionstasten zugeordnet sind. Drückt man auf eine dieser Tasten, erscheint ein Untermenü, daß sich ebenfalls wieder an den Funktionstasten orientiert.

Menügeführt (Funktionstaste F2) kann man die interne Echtzeituhr stellen, beziehungsweise deren Anzeige aufrufen. Mit F3 erreicht man die Zeitschaltuhr. Über die eingebaute Netzsteckdose lassen sich externe Geräte bis 400 Watt Leistungsaufnahme schalten, man kann sich

BMM-BASIC mit Standard-BASIC-Befehlssatz:

BEEP	DIM	NEW
CALL	END	PLIST
CLEAR	FOR .. NEXT	POKE
CLOAD	GOTO	PRINT
CMERGE	GOSUB ... RETURN	PRINTER
CSAVE	IF	RAD
CONSOLE	INPUT	REM
DELETE	LET	RUN
DEG	LIST	WAIT

Problembezogene Erweiterungen:

DOUT	—	Digitale Ausgabe
DIN	—	Digitale Eingabe
AOUT	—	Analoge Ausgabe
AIN	—	Anlaoge Eingabe
NETZ	—	Steuern der 220-V-Steckdose
GEN	—	Erzeugen einer Impulsfolge
XLES/YLES	—	Für Tracer (in Vorbereitung)

Funktionen:

LN,LOG,SOR,EXP	ABS,INT,RND
SIN,COS,TAN	PEEK
ASN,ACS,ATN	CHR$
	LEFT$,RIGHT$

Tabelle 1

Bild 1. Eine Netzsteckdose für Schaltuhrbetrieb ist eingebaut

aber auch mit einem 1000 Hz Fiepton begnügen.

Mit F6 wird der BMM-BASIC-Interpreter aktiviert. MAX' BASIC kann zwar nicht als üppig bezeichnet werden, aber alle notwendigen Befehle sind vorhanden (Tabelle 1).

Dafür sind seine Editormöglichkeiten erfreulich komfortabel. Es gilt der Grundsatz, daß das, was man auf dem Display sieht (und per RETURN abschickt), auch in den Programmspeicher übernommen wird. Diverse Cursor-Funktionen sind vorhanden, auch Insert und Delete ist möglich.

Darüber hinaus erfolgt nach der Eingabe einer jeden Programmzeile ein ausgiebiger Syntax-Check. Der Cursor springt in die unmittelbare Nähe des Fehlers, und der Rechner weigert sich strikt, eine syntaktisch falsche Zeile anzunehmen.

MAXens BASIC hält aber auch Überraschendes bereit: es verfügt über ein paar spezielle Befehle für die bereits eingangs angedeuteten Steuerungsaufgaben.

Unmittelbare Freude hat man als Besitzer der Grundversion zunächst aber nur am Befehl NETZ und GEN. Mit ersterem schaltet man die Steckdose in den jeweils anderen Zustand um, mit GEN kann man Rechtecksignale erzeugen (dieser Befehl ist auch über das Hauptmenü (F4) zugänglich). Unter Angabe von Periodendauer und Anzahl der Schwingungen darf man das Ergebnis über den eingebauten Piezowandler live erleben und kann es über Pin 5 des Kassetten-Interfaces abgreifen.

Wer mehr steuern, regeln oder messen will, der muß nunmehr zu den Steckmodulen greifen (und zur Brieftasche, versteht sich).

(Ein-)Schubweise

Zum Test verfügten wir über folgende Steckmodule:

— DIN/DOUT
Acht Relais, einzeln als Ausgänge ansprechbar. Die Schaltkontakte sind über eine Canon-Buchse herausgeführt. Acht digitale Eingänge.

— AIN
Acht analoge Eingangskanäle (mit AD 7581) über Canon-Buchse.

— AOUT
Zwei analoge Ausgabekanäle (mit ZN 426) über Canon-Buchse.

— EPROM
Zwei leere, aber beschaltete Stecksockel für EPROMs.

Erst mit diesen Modulen kommen die Vorzüge des MAX und seiner steuerungsspezifischen BASIC-Befehle zum Tragen. Normalerweise erfordern auch so einfache Aufgaben, wie das Abfragen von einzelnen Leitungszuständen (bitweise!) oder das Betätigen von Relais, recht umfangreiche Maschinenprogramme. In BMM-BASIC braucht man nur einen Befehl. Mit

DOUT Kanalnummer: Schaltzustand

ist alles bereits erledigt. Mit

DIN Kanalnummer: Numerische Variable

holt man sich den Wert einer Eingangsleitung in eine Varia-

ble. Jede Variable repräsentiert dann allerdings nur ein Bit.

Sehr einfach lassen sich auch die A/D- beziehungsweise D/A-Wandler ansprechen. Mit

AOUT Kanalnummer:
Numerischer Ausdruck

lassen sich beliebige 8-Bit-Werte ausgeben. Wenn in einer Schleife, die Sinuswerte errechnet, jeweils eine Analogausgabe eingebaut wird, verfügt man über einen programmierbaren Sinusgenerator.

Der achtkanalige Analogeingang läßt auf die gleiche einfache Art die Durchführung von mehreren Meßvorgängen zu. Was es also auch sei, einfache Steuer-, Meß- und Regelprobleme sind mit wenigen BASIC-Befehlen abgetan.

Diese Merkmale machen MAX auch für industrielle Anwendungen interessant, wo er zur Entwicklung von Steuerungen oder zum schnellen Ausprobieren bestimmter Sachverhalte ausgezeichnet geeignet ist. Und selbstverständlich erleichtert seine Konzeption auch gerade dem Anfänger den Umgang mit hardwaremäßigen Anwendungen seines Computers.

Aber nicht nur eitel Freude gab es speziell mit den Steckmodulen. Sie haben keine Verriegelung zum Hauptgehäuse, so daß man beim Abziehen des Canon-Steckers üblicherweise das ganze Modul herauszieht. Bei laufendem Gerät kann das mißliebige Folgen haben. Der Mangel wurde aber herstellerseitig erkannt und wird künftig durch eine Schraube behoben. Auch mit den Führungsschienen für die Module machten wir sogleich Bekanntschaft. Nur auf Lack geklebt, fielen uns gleich zwei in die Hände. Im Kunststoffgehäuse sicher kein Problem, beim Metallgehäuse sind aber Verschraubungen nötig.

MAX ist noch sehr jung, er wird noch wachsen. Seine Produktion läuft gerade ein halbes Jahr. Alles, auch die gesamte Betriebssoftware, ist eine Eigenentwicklung der Herstellerfirma. Inzwischen wird bereits ein Video-Interface angeboten, und ein Digitizer ist in Vorbereitung. Was man natürlich schmerzlich vermißt und was auch ganz sicher bald angeboten werden wird, ist ein EPROM-Programmierer.

Die Selbstbauseminare sind zweifellos eine bemerkenswerte Neuerung. Sie gehören inzwischen obligat zum Konzept des MAX. Das heißt ganz konkret: MAX gibt es nur fertig oder als Bausatz einschließlich Seminar. Der 'Zwangs'-Charakter dieser Seminare wird zwar nicht jeden Selbstbauer erfreuen, aber viele Probleme eines Bausatzanbieters, wie Rücksendungen angeblich defekter Bauteile, treten gar nicht erst auf.

Schulung und Hilfestellung

Das Handbuch von MAX umfaßt derzeit knapp 70 DIN-A4-Seiten Text. Nach acht Seiten 'Erste Schritte' folgen gut 20 Seiten vertiefende Informationen über den Rechner und den Umgang mit ihm. Weitere 16 Seiten beschreiben das BMM-BASIC (jeder Befehl wird einzeln mit mehreren Beispielen erklärt), und auf 15 Seiten wird auf die Steckmodule eingegangen. Dazu sind vier konkrete Anwendungsbeispiele angegeben, wie beispielsweise eine Innenwiderstandsbestimmung.

Bestückungspläne und Steckerbeziehungsweise IC-Anschlußbelegungen sind derzeit noch in 'ASCII-Grafik' dargestellt, damit sie (laut Hersteller) per Textverarbeitungssystem leicht änderbar bleiben. Gelegentlich geht das etwas zu Lasten der Übersichtlichkeit.

Lästiger aber sind so kleine Unterlassungssünden wie beim GEN-Befehl, wo man bei Angabe einer Periodendauer von unterhalb einer Millisekunde zwar die Meldung erhält 'Generator läuft', der Piezo aber schweigt. Man könnte ja meinen, daß die Ausgangsleitung Signal führt, und hier nur der Schallwandler (und die Nerven des GENerierers) geschont werden sollen. Aber dem ist nicht so. Folglich lassen sich wohl keine Schwingungen oberhalb 1000 Hz erzeugen.

Solche Probleme mit der Dokumentation werden vermutlich durch die Seminare weitgehend aufgefangen, über deren Qualität wir allerdings keine Aussagen machen können, da wir nur über ein Fertiggerät verfügten. Aber schließlich wird ja garantiert, daß man nicht ohne funktionierendes Gerät nach Hause gehen muß.

Wirklich ärgerlich ist die bewußt betriebene 'Verschleierungstaktik' bezüglich der Schaltpläne. Da immerhin alle Schnittstellen mit Pin-Belegungen von Steckern und Port-ICs 'preisgegeben' werden, könnte sich der Fertiggeräte-Besitzer viele Anfragen sparen, wenn zumindest ein Blockschaltbild mit eben diesen Schnittstellenbezeichnungen vorhanden wäre.

Auf Anfragen, warum man denn einem Selbstbau-Computer kein Schaltbild beilege, wurden Bedenken wegen Raubkopien à la Apple geäußert (was von Selbstbewußtsein kündet). Ausdrücklich ausgeschlossen wurde eine künftige Veröffentlichung allerdings nicht.

Was man anlegen muß

Die abgemagerte Grundversion (4K RAM, geringfügig verringertes Betriebssystem/BASIC, kein Centronics-Stecker und kein Akku, Kunststoffgehäuse) kostet einschließlich Selbstbauseminar unter 800 D-Mark. Über die Hälfte davon machen (laut Herstellerangabe) LC-Display, Schreibmaschinentastatur, Gehäuse und Seminar aus. Zumal MAX nicht aus fernöstlicher Massenproduktion stammt, kann man den Preis wohl als angemessen bezeichnen.

Rund 1000 D-Mark kostet die voll ausgestattete Selbstbauversion (ohne Steckmodule), ebenfalls im Kunststoffgehäuse. Etwa 1900 D-Mark muß man allerdings hinblättern, will man die uns vorliegende Version komplett mit Steckmodulen erwerben. Und es ist kein Seminar im Preis enthalten.

Fazit

Wer sich als Einsteiger in die Computerei einen MAX zulegt, verfügt über ein Gerät mit komfortabler Tastatur, magerem, allerdings auch narrensicherem BASIC und — wenn es gar nicht klappt mit dem Computern — über eine intelligente Digitaluhr mit Schaltvorrichtung.

Aber auch Profis brauchen sich nicht zu genieren, einen MAX zu erwerben. Wenn auch manchen eher maschinensprache-orientierten Interessenten der Prozessor 'zu exotisch' sein mag, um sich darauf einzuarbeiten: für die meisten Probleme wird man mit dem Steuerungs-BASIC auskommen.

Dennoch, ein Bausatzgerät, zu dem kein Schaltplan geliefert wird, stellt schon ein Kuriosum dar. Nicht nur, daß es das grundsätzliche Verständnis der Hardware erschwert, auch eventuell anfallende Reparaturen, die beim Basteln nicht ausbleiben werden, sind fast unmöglich oder müssen vom Hersteller (Kosten!) durchgeführt werden.

Die kleinen Fehler an der Mechanik sind zwar unerfreulich, lassen aber ganz sicher keine Rückschlüsse auf gravierende konzeptionelle Fehler zu. Der optische Eindruck von Innenleben und Gehäuse ist ansonsten ausgesprochen positiv.

Man sollte sich allerdings genau überlegen, ob sich die Anschaffung des Fertiggerätes lohnt: Erstens ist es beachtenswert teurer als die Selbstbauversionen, zum anderen bietet das Seminar eine derzeit wohl noch unverzichtbare Ergänzung zur Dokumentation.

Wenn MAX all das hält, was sich sein Hersteller von ihm verspricht, wird dieser Beitrag sicher nicht der letzte über ihn bleiben. In diesem Sinne: MAX vobiscum! □

Ergebnisse auf einen Blick:

⊕ Ergonomische deutsche Tastatur	⊖ Kein Schaltbild für Selbstbauer
⊕ Insgesamt solide verarbeitet	⊖ Lücken im Handbuch
⊕ Selbstbau-Unterstützung durch Seminar	⊖ Schlecht verklebte Modulführungsschienen
⊕ Sehr einfach erweiterbar (Steckmodule)	⊖ Module bisher noch nicht verriegelt
⊕ Spezielle BASIC-Befehle für Steueraufgaben	⊖ Mageres BASIC
⊕ Komfortabler Editor, gründlicher Syntax-Check	

NEU

HX-20 & CP/M®

Auch in Farbe!

HX-20-Video-Adapter jetzt

die komfortable Verbindung zum Monitor!

7 x 10 Punkt-Matrix, gestochen scharfe Anzeige mit Unterlängen. **Visueller Bildschirm:** 80 Zeilen x 24 Zeilen. **Alle nationalen Zeichensätze** wie HX 20 (alle Editierfunktionen).

STOP

Kompletter HX-20-Zeichensatz (incl. Grafikz. + zusätzl. Zeichen), sämtliche Steuerbefehle, umschaltbar per Programm und Tastatur. Nahezu alle Programme am Monitor ohne Änderung lauffähig.

HX-20-Floppy-Set (bis 1,6 MB)

1—2 Laufwerke, je 380—760 KB, voller HX-20-Befehlssatz, Video-Adapter und Floppy in gleichem oder separatem Gehäuse. CP/M®-Betriebssystem, zusätzlich CP/M®-Programme einsetzbar.

CP/M ist ein Warenzeichen der Digital Research, Inc.

time-soft-EDV®

Sophienstraße 32 · 7000 Stuttgart 1 · Telefon: 0711/22 84 71/72
Programme + Computer für zeitgemäße Anwendungen

Columbia/IBM-PC/Sirius/Victor/Wang

Die

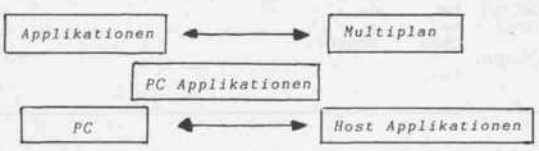
UNIVERSELLE

Software

SCHNITTSTELLE

für PCs

FILE MASTER ermöglicht den problemlosen Datentransfer zwischen



Einfachste Bedienung dank der integrierten Bildschirm-Benutzerführung.
Deutscher Dialog und Dokumentation.

H+I Informatik Aktiengesellschaft
Schwefelstraße 33, FL-9490 Vaduz, Telefon 075/24294

MICROCOMPUTER

ATARI 600XL
ATARI 800XL

MÜNZENLOHER GMBH

Für dieses Computersystem gibt es bereits heute über 2000 Programme !

Wir sind autorisierter Händler für die gesamte ATARI-Computer-Palette. Guter Geräteservice vorhanden. Groß-Servicestelle v. Atari in nächster Nähe.

Tölzer Straße 5
D–8150 Holzkirchen / Obb.
Telefon: (0 80 24) 18 14
Direkt an der Autobahn München - Salzburg.
Gerätebau, Computersysteme – SOFTWARE
Lieferung per NN od. Vorkasse a. Postscheck-Kto. 2845 58–807 München od. Euroscheck.

Die neue SUPERPALETTE von ATARI

ATARI 600XL, 16K RAM, 24K ROM, (inkl. ATARI-BASIC), Netzgerät, dt. Dokumentation **549,– DM**

ATARI 64 MODUL, Speichererweiterung f. Atari 600XL **349,– DM**

Technische Daten wie unten.

ATARI 800XL, 64K RAM, inkl. Atari-Basic, Netzgerät, dt. Beschreibung **899,– DM**

Technische Daten:
24K ROM, eingeb. BASIC, 256 Farben, (davon 16 in BASIC gleichzeitig darstellbar), 320 x 192 Bildpunkte, 24 Zeilen mit 40 Zeichen, 5 Textmodes und 11 Graphikmodes, 4 voneinander unabhängige Tonkanäle (Tonumfang: 3 1/2 Oktaven), Schreibmaschinentastatur mit 4 speziellen Funktionstasten, internationaler Zeichensatz (dt. Umlaute auf Bildschirm sichtbar) mit Graphikzeichen, Help-Taste für Rückfragen und Zwischentitel in verschiedenen Programmen. CPU: 6502 C, 3 spezielle Prozessoren für Graphik, Ton und Bild !

ATARI 1050 Diskettenstation 999,– DM
Bis 127 kB pro Diskette (ca. 100 Schreibmaschinenseiten). Dt./engl. DOS II, Gutschein für DOS III. Wird Betriebsbereit geliefert (Netzgerät, Datenkab., dt. Beschr.) Anschluß über eingebautes Interface.

Lernset 1010 (incl. Recorder, 2 Lernprogr., kompl. mit Anschluß (Netzteil, Kabel) **249,–**

ATARI 1027 Drucker 899,– DM
Schönschreibdrucker mit Interface & Kabel

ATARI 1025 Matrixdrucker 1.399,– DM
Incl. Kabel, Netzteil, Interface

ATARI 1020 Farbdrucker 799,– DM
Dieser Drucker ermöglicht Superfarbgraphik. Incl. Interface, Kabel, Netzt., Progr. Cass.,

Track-Ball f. schnelles präzises Spielen 179,–

AUS UNSERER BÜCHERECKE

ATARI BASIC (deutsch)
Das ideale Buch für den Computer-Neuling m. vielen Programmbeispielen. **39,– DM**

Das große Spielebuch für ATARI 600/800 XL
Ca. 20 Spiele **nur 29,80**

Mein ATARI-Computer 56,– DM
Das beste Buch über den ATARI! Der ATARI wird von allen Seiten her durchleuchtet. Hardware, Peripherie, BASIC-Einführung, fortgeschrittene Programmiertechniken, Graphik, Sound u. vieles mehr. Auch neue XL-Typen werden behandelt!

Die Trickkiste 39,– DM
Neue Tips, Tricks und Programme für ATARI

ATARI BASIC spielend lernen 10,80 DM

101 ATARI Computer Programming 29,– DM
Tips & Tricks

31 New ATARI Computer Programs 29,– DM
2 Bücher m. einer Vielzahl interessanter Progr.

Einführung in LOGO 42,– DM
Eine Programmiersprache speziell für Graphikdarstellung auf dem Bildschirm (bald auch für ATARI erhältlich).

Der ATARI-Assembler 36,– DM
Deutschsprachiges Buch über Assemblerprogrammierung auf dem ATARI.

The Best of Softside 76,– DM
Die besten Programme aus dem Computermagazin "Softside".

ATARI Sound und Graphics 49,– DM
Sound und Graphik für jeden ATARI-User

Mapping the ATARI 58,– DM
Alle wichtigen Adressen und Vektoren Ihres ATARI-Computers. Eines der wichtigsten Bücher um den Computer voll zu beherrschen.

ATARI Graphics 49,– DM
Viel wissenswertes über Graphikprogrammierung auf dem ATARI.

Machinelanguage for Beginners 49,– DM
Hervorragendes Buch zum Erlernen der 6502-Maschinensprache (350 Seiten !).

ATEXT – Wortprozessor f. ATARI mit dt. Anleitung. Einer der besten Wortprozessoren weltweit! (C) 148,– DM
(D) 159,– DM (ROM-Modul) 199,– DM

NEU — SHARP MZ-731 — NEU

Technische Daten:
Z-80A (3,5 MHz), 64 KB, ASCII-Tastatur, Cursor-Steuertasten, Definierbare Funktionstasten, Tasten zum Einfügen und Löschen (INS, DEL), Verwendung eines Fernsehgerätes (PAL-System) oder einer Monitoranzeige möglich. RGB-Anschluß, Lautsprecher u. Uhr-Funktion eingebaut, zusätzlicher Drucker- und Floppy-Disk-Anschluß möglich. Mit Sharp BASIC Interpreter, mit 2 Cassetten u. ausf. dt. Handbuch. Mit Cassetten-Recorder Vierfarbdrucker u. 10 Progr. **nur 1395,–**

NEUE BÜCHER
Schritt für Schritt mit MZ-700
von Prof. Dr. Günther O. Hamann 29,80
Programmier-Handbuch für SHARP 49,–

Programme für SHARP MZ-700

PCG-Graphiksystem (Modul) **360,– DM**
Erlaubt Ansteuerung aller Bildschirmrasterpunkte und Neudef. des Zeichensatzes.

Mini HUBASIC-Compiler (C) **120,– DM**
Urtext-700 (C) **120,– DM**
Sehr komfortables Textverarbeitungsprogr.

Urtab (C) **120,– DM**
Flexibles Datenverwaltungsprogramm

FORTRAN-Compiler (C) **95,– DM**
Programmausf. 50 mal schneller als in BASIC

Superdisassembler (C) **75,– DM**
Super-ML C000 bzw. 5000 (C) **55,– DM**
Zum Programmieren in Maschinensprache

Adressverwaltung (C) **98,– DM**
Lagerverwaltung (C) **99,– DM**
Terminverwaltung (C) **99,– DM**

Börsenprogramm - wurde v. einen Börsenfachmann entwickelt u. erstellt. Es ermöglicht Ihnen Kauf- u. Verkaufstermine von Aktien zu bestimmen. Sie geben laufend d. aktuellen Aktienkurse ein. Der Rechner wertet diese Daten aus, druckt Graphiken aus, liefert Trendmeldungen, Kauf- u. Verkaustermine u. sortiert alle eingegebenen Aktien nach verschiedenen Kriterien. **99,– DM**

Über 100 Spielprogramme zwischen 20,– und 100,– DM f. d. SHARP stehen zur Verfügung.

HARDWARE
Original SHARP-Laufwerke m. 1 od. 2 Laufwerken, 5 1/4 Zoll, Doppelseitig, doppelte Schreibdichte, je 280 kB mit Betriebssystem.
Formschöner **SHARP-Monitor MZ-1D05,**
RGB-Eingang m. brillianten Farbbild **1.425,–**

Die 2 speziellen Hefte f. ATARI (ANTIC u. ANALOG) sowie COMPUTE a' **15,80 DM**

BUSIPACK
Das Super-Geschäftsprogramm für den Vertriebsmann u. den Kleinbetrieb. Lager- (1000 Artikel) u. Adressenveraltung (500 Adressen). Auf Wunsch 1000 Adressen u. 500 Artikel. Rechnungsschreib-Progr., Lager wird mitgeführt mit Bestellwesen f. ATARI **499,– DM**

Bücher für den C-64

Wirtschaft auf dem C-64 38,– DM
40 Progr. aus d. Bereichen Finanzmathem., Unternehmensorschung u. Betriebswirtschaft.

Grafik auf dem Commodore 64 38,– DM

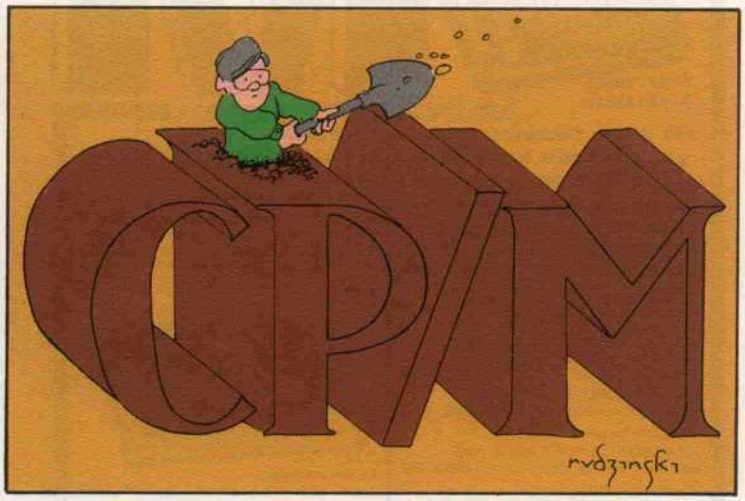
BASIC auf dem Commodore 64 38,– DM
BASIC-Einführung u. Erläuterung spezifischer Eigenschaften.

Beherrschen Sie Ihren Commodore 64 19,80
64 Programme für den C-64 39,–
Hardware Erweiterung f. C-64 39,–
Progr. i. Maschinensprache m. C-64 29,80
Mehr als 29 Programme f. C-64 29,80

NEU – BLIZTEXT 1.1 für C-64 199,–

NEU – BUSIPACK f. C-64 – Lager-, Adressenverw. u. Fakturierung 299,– DM

Fordern Sie noch heute unseren Katalog speziell für den SHARP bzw. ATARI an! Wir freuen uns auch auf Ihren Besuch.



Das Betriebssystem CP/M-86

Anpassung, Teil 2

Kurt Werner

Im ersten Teil dieses Artikels wurde gezeigt, wie das BIOS des CP/M-86 an die Hardware des c't 86 anzupassen ist. Damit sind die Vorarbeiten für den ersten Praxistest abgeschlossen. Dieser Artikel zeigt, wie ein lauffähiges Betriebssystem erstellt werden kann. Die dazu notwendigen Hilfsprogramme werden ebenfalls beschrieben.

Neben dem angepaßten BIOS sind zur Erstellung einer bootfähigen Diskette noch drei Hilfsprogramme nötig: Ein Urlader, der das eigentliche CP/M von der Diskette in den Speicher lädt, ein Programm zum Kopieren der Systemspuren einer Diskette und ein Formatierprogramm, das eine Diskette erst beschreibbar macht. Mit diesen Programmen kann man eine Diskette mit einem lauffähigen CP/M-86 erstellen.

Der Urlader

Dieses Ladeprogramm steht auf der ersten Spur der Diskette. Beim Bootvorgang sorgt der Lader dafür, daß die ersten beiden Spuren gelesen werden. Dazu muß das Programm an das verwendete Laufwerksformat angepaßt sein. Die Definition des Programms kann für 5¼"-Drives und für 8"-Laufwerke vorgenommen werden.

Soll der Urlader für ein System mit 5¼"-Laufwerken generiert werden, ist im Source-Listing die Variable 'FUENF' auf 'TRUE' zu setzen. Für 8"-Drives muß die Variable den Wert 'FALSE' erhalten.

Die zweite Anpassung des Programms an die Laufwerke betrifft die Aufzeichnungsformate. Das Programm lädt immer die ersten beiden Spuren der Diskette in den Speicher, egal, ob es sich um ein einseitiges oder doppelseitiges Laufwerk handelt. Dieses Format kann durch den Wert der Variablen 'MAXSEC' bestimmt werden. Dabei wird lediglich die Anzahl der maximal verfügbaren Sektoren festgelegt: Bei 8"-Laufwerken (einseitig) 27 und bei zweiseitigen 8"-Drives 53. Diese Zahlen stellen die reale Sektoranzahl minus eins (bedingt durch eine Inkrementation der Sektorzahl beim Programmaufruf) dar. Für 5¼"-Drives muß der Urlader an das Format des 'Boot-Laufwerks' angepaßt werden. Normalerweise kann bei einseitigen Laufwerken die Variable 'MAXSEC' auf 11, und bei doppelseitigen Drives auf 21 gesetzt sein.

Um den Lader mit den korrekten Laufwerksangaben zu generieren, muß folgende Befehlsfolge in den Rechner eingegeben werden:

```
ASM86 BOOT $PX
GENCMD BOOT
```

```
;
; B O O T   BOOTER FUER CP/M-86
;
TRUE    EQU     -1
FALSE   EQU     0
;
FUENF   EQU     TRUE
ACHT    EQU     NOT FUENF
;
; MONITORAUFRUFE
;
HOME    EQU     0AH
SETDMA  EQU     HOME+2
SETTRK  EQU     SETDMA+2
SETDRV  EQU     SETTRK+2
SETSEC  EQU     SETDRV+2
SEDMAS  EQU     SETSEC+2
READ    EQU     SEDMAS+2
;
        CSEG    0F000H
        ORG     0E000H
MONITOR:
;
CPMSTN  EQU     0810H   ;STARTADRESSE VOM LOADER CP/M-86
BOOTER  EQU     08000H  ;STARTADRESSE DES INITIAL BOOTERS
CPMSTS  EQU     0H      ;STARTSEGMENT VOM CP/M-86
MAXTRK  EQU     2H      ;MAXIMALZAHL DER TRACKS
;
        IF      FUENF
CPMLD   EQU     8200H   ;STARTADRESSE DES ZU LADENDEN CODES
OFSET   EQU     200H    ;512 BYTES / SEKTOR
MAXSEC  EQU     21      ;20 SEKTOREN AUF DER BOOTDISKETTE
        ENDIF
;
        IF      ACHT
CPMLD   EQU     80B0H
OFSET   EQU     80H
MAXSEC  EQU     1BH
        ENDIF
;
; START DES CODES
;
        CSEG    0
        ORG     0
;
        MOV     AX,CS
        MOV     DS,AX           ;DATENSEGMENT = CODESEGMENT
        MOV     ADRESS,CPMLD    ;INITIIEREN
; DMAADRESSE SETZEN
        MOV     CX,CPMSTS
        MOV     AL,SEDMAS
        CALLF   MONITOR
;
; DIE ANDEREN PARAMETER SIND SCHON AUS
; DEM EPROM HERAUS GESETZT
;
        MOV     TRACK,0         ;BEGIN : TRACK 0,SEKTOR 2
        MOV     SECTOR,2H
        JMPS    BOOTBE
BOOTL:  MOV     CX,SECTOR
        MOV     AL,SETSEC
        CALLF   MONITOR         ;SEKTOR SETZEN
        MOV     CX,ADRESS
        MOV     AL,SETDMA
        CALLF   MONITOR         ;UEBERTAGUNGSBEREICH SETZEN
        MOV     AL,READ
        CALLF   MONITOR         ;LESEN
        ADD     ADRESS,OFSET    ;NAECHSTEN UEBERTRAGUNGSBEREICH
                                 FESTLEGEN
        INC     SECTOR
        CMP     SECTOR,MAXSEC   ;LEZTER SEKTOR ?
        JNZ     BOOTL
        INC     TRACK           ;LETZTER TRACK ?
        CMP     TRACK,MAXTRK
        JZ      BOOTEN
        MOV     SECTOR,1        ;NAECHSTEN TRACK SETZEN
BOOTBE: MOV     CX,TRACK
        MOV     AL,SETTRK
        CALLF   MONITOR
        JMPS    BOOTL
BOOTEN: JMPF    LOADER
DATA    EQU     OFFSET $
;
        DSEG
        ORG     DATA
ADRESS  DW      0
SECTOR  DW      0
TRACK   DW      0
        CSEG    CPMSTN
LOADER:
        END
```

SCOPY-Programm

Dieses Programm ermöglicht es, Dateien auf die Systemspuren einer Diskette zu kopieren. SCOPY entspricht somit dem SYSGEN-Programm des CP/M-80. Da der c't 86 auch mit 5¼"-Laufwerken betrieben werden kann, ist das Programm 'LDCOPY' von DIGITAL RESEARCH nicht verwendbar: LDCOPY funktioniert nur bei 8"-Drives.

Das Programm 'SCOPY' muß, bevor es eingesetzt wird, an die

```
;
; SYSTEMCOPY
;
TRUE    EQU     -1
FALSE   EQU     NOT TRUE
;
FUENF   EQU     TRUE
ACHT    EQU     TRUE
;
        CSEG    0
        ORG     05CH
;
; DEFINITION DES UEBERGEBENEN PARAMETERS
;
FCB:
;
; BEGINN DES PROGRAMMS
;
        ORG     0100H
; DS=CS
        MOV     AX,CS
        MOV     DS,AX
        MOV     DX,OFFSET START
        MOV     CL,9
        INT     224
        MOV     WRITE,0
        MOV     BX,OFFSET FCB
        MOV     AL,1[BX]
; PARAMETER VORHANDEN ?
        CMP     AL,20H
        JZ      READTO
; JA, EINLESEN EINES FILES
      - CALL    READFILE
LOOP:   MOV     CL,9
;
; SCHREIBEN AUF DISKETTE
;
        MOV     DX,OFFSET DESTW
        INT     224
        MOV     CL,1
        INT     224
        CMP     AL,03H
        JNZ     NOABORT
; ENDE
FINI:   MOV     DL,0
        MOV     CL,0
        INT     224
NOABORT:        CMP     AL,46H
        JA      LOOP
        CMP     AL,41H
        JB      LOOP
        SUB     AL,41H
        MOV     DRIVE,AL
        MOV     WRITE,1
; AUF AUSGEWAEHLTEM DRIVE SCHREIBEN
        CALL    WRITETO
        JMP     LOOP
;
READTO: CALL    READT
        JMP     LOOP
;
; SCHREIBEN ODER LESEN AUF SYSTEMSPUREN
;
READT:  MOV     DX,OFFSET SOURCE
        MOV     CL,9
        INT     224
; DRIVE EINLESEN
        MOV     CL,1
        INT     224
        CMP     AL,3
        JNZ     NOAB
        MOV     CL,0
        MOV     DL,0
        INT     224
NOAB:   CMP     AL,41H
        JB      READTO
        CMP     AL,46H
        JA      READTO
        SUB     AL,41H
        MOV     DRIVE,AL
;BERECHNUNG DER KLEINSTEN UND DER GROESSTEN SPUR
WRITETO:        MOV     AL,DRIVE
        ADD     AL,AL
        MOV     BX,OFFSET DRIVETABLE
        MOV     AH,0
        ADD     BX,AX
        MOV     CX,[BX]
        MOV     MINSEC,CL
        MOV     CURSEC,CL
        MOV     MAXSEC,CH
;LAUFWERK SETZEN
        MOV     CL,DRIVE
        MOV     AL,9
        CALL    BIOS
        MOV     CURTRACK,0
        MOV     AL,8
;HOME
        CALL    BIOS
        MOV     CX,CS
        MOV     AL,17
;DATENSEGMENT IST DAS CURRENT CODE SEGMENT
        CALL    BIOS
        MOV     AX,OFFSET DATEN+80H
        MOV     ADRESS,AX
READT1: MOV     AL,11
        MOV     CL,CURSEC
;SEKTOR AUSGEBEN
        CALL    BIOS
        MOV     CX,ADRESS
        MOV     AX,12
;DMA-ADRESSE AUSGEBEN
        CALL    BIOS
        MOV     AL,13
;LESEN ODER SCHREIBEN
        CMP     WRITE,0
        JZ      READT2
        MOV     AL,14
READT2: MOV     CL,0
        CALL    BIOS
        OR      AL,AL
        JZ      NOERROR
;FEHLER BEIM LESEN ODER SCHREIBEN
NEXTC:  MOV     CL,9
        MOV     DX,OFFSET ERROR
        INT     224
        MOV     CL,1
        INT     224
        CMP     AL,'N'
        JNZ     M2
        JMP     FINI
M2:     CMP     AL,'J'
        JNZ     NEXTC
NOERROR:        ADD     ADRESS,80H
        INC     CURSEC
        MOV     AL,CURSEC
        CMP     AL,MAXSEC
        JLE     READT1
;ZWEITE SPUR ?
        CMP     CURTRACK,0
        JNZ     RETUR
        INC     CURTRACK
        MOV     CX,1
        MOV     AL,10
; SPUR 1 SETZEN
        CALL    BIOS
        MOV     AL,MINSEC
        MOV     CURSEC,AL
        JMP     READT1
RETUR:  RET
;
; READ SYSTEM FORM FILE
;
READFILE:       MOV     CL,15
        MOV     DX,OFFSET FCB
        MOV     BX,9
; FILE EXTENSION = CMD
        MOV     FCB[BX],43H
        INC     BX
        MOV     FCB[BX],4DH
        INC     BX
        MOV     FCB[BX],44H
; OPEN FILE
        INT     224
        CMP     AL,255
        JZ      FND
        MOV     CL,51
        MOV     DX,CS
; DMA-SEGMENT-ADRESSE = CS
        INT     224
        MOV     AX,OFFSET DATEN
        MOV     ADRESS,AX
READB:  MOV     DX,ADRESS
; DMA-ADRESSE SETZEN
        MOV     CL,26
        INT     224
        MOV     CL,20
        MOV     DX,OFFSET FCB
; LESEN
        INT     224
        OR      AL,AL
        JNZ     READE
        ADD     ADRESS,80H
        JMP     READB
; FILE ENDE
READE:  MOV     CL,16
        MOV     DX,OFFSET FCB
; CLOSE FILE
        INT     224
        RET
;
; ERROR !  FILE NICHT DA
;
FND:    MOV     DX,OFFSET FILENICHTDA
        MOV     CL,9
        INT     224
; AUSGANG ZUM BETRIEBSSYSTEM
        MOV     DL,0
        MOV     CL,0
        INT     224
;
; AUSGABE AN DAS BIOS
;
BIOS:   MOV     TABLE,AL
        MOV     CXDES,CX
        MOV     DXDES,DX
        MOV     CL,50
        MOV     DX,OFFSET TABLE
        INT     224
        RET
;
; DATENSEGMENT
;
DATA    EQU     OFFSET $
```

```
        DSEG
        ORG     DATA
DRIVE   DB      0
CURSEC  DB      0
MINSEC  DB      0
MAXSEC  DB      0
OFF     DW      0
CURTRACK        DB      0
WRITE   DB      0
TABLE   DB      0
CXDES   DW      0
DXDES   DW      0
ADRESS  DW      0
DRIVETABLE      RW      0
;
; MINIMALE UND MAXIMALE SEKTOREN
;
        IF      ACHT
        DB      1,1AH
        DB      1,1AH
        ENDIF
;
        IF      FUENF
        DB      0,79
        DB      0,79
        DB      0,63
        DB      0,63
        ENDIF
;
START   DB      13,10
        DB      'SYSTEMGENERIERUNG VERSION 1.0'
        DB      13,10,'$'
SOURCE  DB      'BITTE SOURCEDRIVE AUSWAEHLEN (A-F)'
        DB      13,10,'$'
DESTW   DB      13,10
        DB      'BITTE DESTINATIONDRIVE AUSWAEHLEN (A-F)'
        DB      13,10,'$'
FILENICHTDA     DB      'FILE IST NICHT AUF DER DISKETTE'
        DB      13,10,'$'
ERROR   DB      'SCHREIB- ODER LESEFEHLER. WEITER (J/N) ?'
        DB      13,10,'$'
DATEN   RS      5100H
        DB      0
        END
```

vorhandene Systemkonfiguration angepaßt werden. Dazu sind im Source-Listing die Variablen 'FUENF' und 'ACHT' dem System entsprechend zu setzen (siehe auch Urlader). Danach ist die Drivetabelle an das System anzupassen. In der Tabelle stehen jeweils die kleinste und die größte Sektornummer, auf der das Betriebssystem steht. Die angegebenen Werte gelten für zweiseitige 5¼"-Laufwerke und einseitige 8"-Drives. Bei anderen Laufwerken sind die Werte durch zwei zu teilen (5¼"-Drives), beziehungsweise zu verdoppeln (8"-Drives). Nach der Anpassung muß das Programm assembliert werden. Dazu dient folgende Befehlsfolge:

ASM86 SCOPY $PX
GENCMD SCOPY 8080

Danach steht das lauffähige Programm bereit.

SCOPY bietet nun zwei Möglichkeiten, den zu kopierenden File zu bestimmen. Zum einen kann beim Programmaufruf der Name einer Datei angegeben werden. Es muß sich dabei um eine Systemdatei mit der Extension '.CMD' handeln. Diese Datei wird dann eingelesen und anschließend in die ersten beiden Spuren der Diskette geschrieben. Dieser Vorgang ist notwendig, wenn ein neues Betriebssystem generiert werden soll. Verfügt man über ein lauffähiges Betriebssystem auf einer Diskette, kann es eingelesen und dann in die Spuren einer anderen Diskette kopiert werden. Um diesen Vorgang auszulösen reicht es aus, SCOPY ohne Angabe eines Parameters aufzurufen. Das Programm fragt dann nach dem Laufwerk von dem kopiert werden soll, anschließend nach dem Bestimmungslaufwerk. Dabei kann der Programmlauf jederzeit durch die Eingabe von 'Controll C' abgebrochen werden.

FORMAT

Da wohl die Mehrzahl der c't-86-Interessierten 5¼"-Laufwerke einsetzen wird, stellen wir ein Formatierprogramm für dieses Floppyformat vor. Bei 'FORMAT' entfällt zwar die Auswahl des verwendeten Laufwerktyps, eine Anpassung an das System muß trotzdem erfolgen. Dazu sind die Variablen SINGLE, DOUBLE, SECMAX, TRKMAX und STEP entsprechend dem System zu setzen. SINGLE und DOUBLE bestimmen, ob einseitige oder zweiseitige Laufwerke bedient werden sollen. Zur Anpassung ist die entsprechende Variable auf den Wert 'TRUE' zu setzen, die andere auf den Wert 'FALSE'.

SECMAX gibt die Anzahl der

```
;
; FORMATIERPROGRAMM
;
TRUE    EQU     -1
FALSE   EQU     NOT TRUE
;
SINGLE  EQU     FALSE
DOUBLE  EQU     NOT SINGLE
;
; KONSTANTEN, SYSTEMABHAENGIG
;
SECMAX  EQU     10
TRKMAX  EQU     80
STEP    EQU     0
;
;
DREHZAHL        EQU     0B8H
;
; KONSTANTEN, SYSTEMUNABHAENGIG
;
K4E     EQU     04EH
KF6     EQU     0F6H
KF5     EQU     0F5H
KFC     EQU     0FCH
KF7     EQU     0F7H
;
FLOPI   EQU     0BAH
FLOPD   EQU     0B6H
INT2    EQU     03EH
REFPORT EQU     08CH
FLOPO   EQU     0B8H
FLOPC   EQU     080H
;
;
        CSEG
        ORG     0100H
;
START:  MOV     DX,OFFSET STARTMESS
        MOV     CL,9
        INT     224
; AUSGABE DER OPTIONEN
; MAXIMALER SEKTOR
        IF      SECMAX EQ 8
        MOV     DX,OFFSET SEK8
        ENDIF
        IF      SECMAX EQ 9
        MOV     DX,OFFSET SEK9
        ENDIF
        IF      SECMAX EQ 10
        MOV     DX,OFFSET SEK10
        ENDIF
; MAXIMALE TRACK
        MOV     CL,9
        INT     224
        IF      TRKMAX EQ 40
        MOV     DX,OFFSET TR40
        ENDIF
        IF      TRKMAX EQ 80
        MOV     DX,OFFSET TR80
        ENDIF
; ANZAHL DER SEITEN DER DISKETTE
        MOV     CL,9
        INT     224
        IF      SINGLE
        MOV     DX,OFFSET SI1
        ENDIF
        IF      DOUBLE
        MOV     DX,OFFSET SI2
        ENDIF
        MOV     CL,9
        INT     224
        MOV     DX,OFFSET STARTME
        MOV     CL,9
        INT     224
DRIVENOT:       MOV     CL,1
        INT     224
        CMP     AL,3    ;CONTROL C ?
        JNZ     FINI1
        JMP     FINI
; DRIVE ?
FINI1:  CMP     AL,41H
        JNGE    DRIVENOT
        CMP     AL,44H
        JA      DRIVENOT
DRIVEFOUND:     SUB     AL,41H
        MOV     AH,0
        MOV     DRIV,AX
        MOV     BX,AX
        MOV     AL,DRIVES[BX]
        OUT     FLOPO,AL
ANFANG: MOV     DX,OFFSET CRLF
        MOV     CL,9
        INT     224
; HOME
BEGIN:  MOV     AL,8+STEP
        OUT     FLOPC,AL
        MOV     TRACK,0
; SPUR SETZEN
LOOP:   MOV     AL,TRACK
        OUT     FLOPD,AL
        MOV     AL,18H+STEP
        OUT     FLOPC,AL
; SPURNUMMER AN TERMINAL AUSGEBEN
        MOV     AL,TRACK
        MOV     DL,10
        DIV     DL
        PUSH    AX
        CALL    BIHEX
        POP     AX
```

```
        MOV     AL,AH
        CALL    B1HEX
; CURSOR ZURUECKSETZEN
        MOV     CL,2
        MOV     DL,8
        INT     224
        MOV     DL,8
        MOV     CL,2
        INT     224
; FLOPPYCONTROLER FERTIG ?
WAIT1:  IN      AL,FLOPC
        TEST    AL,1H
        JNZ     WAIT1
        CALL    MAKETRACK
        CALL    WRITETRACK
        INC     TRACK
        CMP     TRACK,TRKMAX
        JNZ     LOOP
; ZWEISEITIGE DISKETTE
        IF      DOUBLE
        CMP     SIDE,1
        JZ      ENDER
        MOV     SIDE,1
        JMP     BEGIN
        ENDIF
; ENDE
ENDER:  MOV     DX,OFFSET ENDMESS
        MOV     CL,9
        INT     224
        MOV     SIDE,0
        MOV     CL,1
        INT     224
        CMP     AL,04AH
        JZ      ANFANG
        CMP     AL,'N'
        JNZ     ENDER
FINI:   MOV     CL,0
        MOV     DL,0
        INT     224     ;ENDE
;
; AUSGABE EINER ZIFFER AN TERMINAL
;
B1HEX:  AND     AL,0FH
        CMP     AL,0AH
        JL      B0HEX
        ADD     AL,7
B0HEX:  ADD     AL,30H
        MOV     DL,AL
        MOV     CL,2
        INT     224
        RET
;
;AUFBAUEN EINER SPUR
;FORMATAUFBAU SIEHE DATENBLATT 2797
;
MAKETRACK:      MOV     CURSEC,1
        MOV     BX,0
        MOV     AL,K4E
        MOV     CX,80
        CALL    DUP
        MOV     AL,0
        MOV     CX,12
        CALL    DUP
        MOV     AL,KF6
        MOV     CX,3
        CALL    DUP
        MOV     SEKANF[BX],KFC
        INC     BX
        MOV     AL,K4E
        MOV     CX,50
        CALL    DUP
MLOOP:  MOV     AL,0
        MOV     CX,12
        CALL    DUP
MOV     AX,KF5
MOV     CX,3
CALL    DUP
MOV     SEKANF[BX],0FEH
INC     BX
MOV     AL,TRACK
MOV     SEKANF[BX],AL
INC     BX
MOV     AL,SIDE
MOV     SEKANF[BX],AL
INC     BX
MOV     AL,CURSEC
MOV     SEKANF[BX],AL
INC     BX
MOV     SEKANF[BX],2
INC     BX
MOV     SEKANF[BX],KF7
INC     BX
MOV     AL,04EH
MOV     CX,22
CALL    DUP
MOV     AL,0
MOV     CX,12
CALL    DUP
MOV     AL,0F5H
MOV     CX,3
CALL    DUP
MOV     SEKANF[BX],0FBH
INC     BX
MOV     AL,0E5H
MOV     CX,512
CALL    DUP
MOV     SEKANF[BX],0F7H
        INC     BX
        MOV     AL,04EH
;
; VARIABEL IST DER ABSTAND DER SEKTOREN
; VONEINANDER AUF DER DISKETTE
; 10 SEKTOREN = 26 BYTES
; 9 SEKTOREN = 66 BYTES
; 8 SEKTOREN = 150 BYTES
;
        MOV     CX,22*SECMAX*SECMAX-458*SECMAX+2406
        CALL    DUP
        INC     CURSEC
        CMP     CURSEC,SECMAX
        JLE     MLOOP
        MOV     AL,04EH
        MOV     CX,1598
        CALL    DUP
        RET
;
; DUPLIZIEREN VON SPEICHERBEREICHEN
;
DUP:    MOV     SEKANF[BX],AL
        INC     BX
        LOOP    DUP
        RET
;
WRITETRACK:
; WRITE PROTECTED DISK ?
        IN      AL,FLOPC
        TEST    AL,40H
        JZ      WRY0
        MOV     DX,OFFSET WRITE
        MOV     CL,9
        INT     224
        MOV     CL,0
        INT     224
; DRIVE READY ?
WRY0:   TEST    AL,80H
        JZ      WRY1
        MOV     DX,OFFSET READY
        MOV     CL,9
        INT     224
        MOV     CL,0
        INT     224
; WARTEN AUF INDEXLOCH
WRY1:   IN      AL,FLOPI
        TEST    AL,8
        JNZ     WRY1
        IN      AL,FLOPI        ;ENTPRELLEN
        TEST    AL,8
        JNZ     WRY1
; WARTEN AUF NAECHSTES INDEXLOCH
        MOV     AX,DREHZAHL
WL1:    CALL    WIMS
        DEC     AX
        JNZ     WL1
        MOV     DL,2
        MOV     DH,1
        MOV     BX,DRIV
        MOV     AL,DRIVES[BX]
        LEA     BX,SEKANF
        AND     AL,7FH
        MOV     DI,0
        CMP     SIDE,0
        JZ      NOKORR
        MOV     DI,2
NOKORR: OR      AL,40H
        OUT     FLOPD,AL
        AND     AL,3FH
        OUT     FLOPD,AL
        CLI
        MOV     AL,0FBH
        OUT     INT2,AL
        MOV     AL,0F0H
        STI
        XOR     AX,DI
        OUT     FLOPC,AL
        OUT     REFPORT,AL
WRLOOP: IN      AL,FLOPC
        TEST    AL,DH
        JZ      WREND
        TEST    AL,DL
        JZ      WRLOOP
        MOV     AL,[BX]
        OUT     FLOPD,AL
        INC     BX
        JMPS    WRLOOP
WREND:  PUSH    AX
        PUSH    DS
; INTERRUPTE FREISCHALTEN
        MOV     AX,0F000H
        MOV     DS,AX
        MOV     AL,INTERR
        POP     DS
        OUT     INT2,AL
        OUT     REFPORT,AL
        POP     AX
        TEST    AL,1CH
        JZ      NOERR
        MOV     DX,OFFSET ERROR
        MOV     CL,9
        INT     224
        MOV     AX,80
WAITH:  CALL    WIMS
        DEC     AX
        JNZ     WAITH
        JMP     WRITETRACK
NOERR:  RET
```

```
WIMS:   MOV     CX,290
WIMSA:  LOOP    WIMSA
        RET
ENDE    EQU     OFFSET $
;
        DSEG
        ORG     ENDE
DRIVES  DB      1H      ;DOUBLE DENSITY
        DB      2H      ;DOUBLE DENSITY
        DB      4H
        DB      8H
DRIV    DW      0
TRACK   DB      0
CURSEC  DB      0
SIDE    DB      0
STARTMESS       DB      13,10
        DB      'FORMATIERPROGRAMM FUER 5 1/4 ZOLL DISKETTEN'
        DB      13,10,'$'
SEK8    DB      'ACHT SEKTOREN PRO TRACK'
        DB      13,10,'$'
SEK9    DB      'NEUN SEKTOREN PRO TRACK'
        DB      13,10,'$'
SEK10   DB      'ZEHN SEKTOREN PRO TRACK'
        DB      13,10,'$'
TR40    DB      '40 TRACKS PRO DISKETTE'
        DB      13,10,'$'
TR80    DB      '80 TRACKS PRO DISKETTE'
        DB      13,10,'$'
SI1     DB      'EINSEITIGE DISKETTE'
        DB      13,10,'$'
SI2     DB      'ZWEISEITIGE DISKETTE'
        DB      13,10,'$'
STARTME DB      'DISKETTE EINLEGEN UND LAUFWERK AUSWAEHLEN'
        DB      13,10,'$'
ENDMESS DB      13,10
        DB      'FORMATIERUNG BEENDET. NOCH EINE DISKETTE
                 FORMATIEREN?'
        DB      13,10,'$'
WRITE   DB      13,10
        DB      'DISKETTE SCHREIBGESCHUETZT'
        DB      13,10,'$'
READY   DB      13,10
        DB      'DRIVE NOT READY'
        DB      13,10,'$'
ERROR   DB      13,10
        DB      'SCHREIBFEHLER'
CRLF    DB      13,10,'$'
;
SEKANF  RS      18000
        DB      0
;
; LETZES BYTE IM MONITOR =
; FREIGABEBYTE FUER INTERRUPTE
;
        ORG     0FFFFH
INTERR  RS      0
        END
```

Sektoren je Track an. Dieser Variablen können die Werte Acht, Neun oder Zehn zugeordnet werden.

Die Variable TRKMAX gibt die maximale Anzahl der Spuren auf der Diskette an. Hier können die Werte 40 oder 80 eingetragen werden.

Die letzte zu definierende Variable heißt STEP und gibt die Steprate des Laufwerks an. Die für das verwendete Laufwerk relevanten Werte können der Tabelle 1 entnommen werden (siehe auch c't 2/84, Seite 86, Tabelle 5). In der Tabelle entspricht R1 dem Bit 1, R0 der Variablen. In STEP darf kein Wert eingetragen werden, der größer als drei ist.

Praxis

Sind alle Systemanpassungen vorgenommen, ist der c't 86 einsatzbereit. Doch mancher stolze Besitzer wird Schwierigkeiten haben, die notwendigen Anpassungen der Software selbst vorzunehmen. In diesem Fall bieten sich zwei Auswege an: Zum einen kann ein sachkundiger Bekannter Hilfestellung geben. Neben seinen Fachkenntnissen sollte er auch mindestens über ein lauffähiges CP/M-80-System und einen 8086-Cross-Assembler verfügen. Eine andere Lösung ist, sich eine bootfähige Diskette zu kaufen. Dieser Weg zum Erfolg ist sicher der schnellste und einfachste, aber wohl auch der kostspieligste. Damit c't-86-Anwender trotzdem zu einem preiswerten CP/M-86 kommen, haben wir uns etwas einfallen lassen (siehe Kasten).

Tabelle 1. Step-Raten des Floppy-Controllers

Step-Rate		Zeit in ms	
R1	R0	5¼"	8"
0	0	6	3
0	1	12	6
1	0	20	10
1	1	30	15

Preisgünstig zu CP/M 86

Ein fertig angepaßtes CP/M 86 für den c't 86 ist nicht billig. Das liegt daran, daß die relativ hohen Lizenzkosten bei geringen Stückzahlen voll zu Buche schlagen. Es gibt aber einen Weg, preisgünstig zu einem angepaßten Betriebssystem zu kommen: c't-86-Anwender können eine CP/M-86-Version für den IBM PC kaufen, die im Handel schon für rund 200 Mark erhältlich ist. Beim Software-Service des Heise-Verlags kann man das BIOS für den c't 86 sowie das Boot-Programm, das Formatierungsprogramm und das Programm Systemcopy auf die Original-Diskette kopieren lassen. Ergebnis: Ein CP/M-86-Betriebssystem für den c't 86. Der Kopierdienst kostet 20 Mark.

Wer diesen Service nutzen will, sollte die System-Diskette(n) in ausreichend fester Verpackung an den Heise-Software-Service, Bissendorfer Straße 8, 3000 Hannover 61, senden. Legen Sie bitte einen Verrechnungsscheck über den Betrag von 20 DM und einen Zettel mit genauer Angabe des Diskettenformats bei. Erlaubt sind folgende Formate:

5¼-Zoll, 40 Track, SS oder DS, DD, 8 oder 9 Sektoren pro Track und

8-Zoll, SS, SD (IBM-Standard-Format 3740).

Das Einkopieren der genannten Programme erfolgt ohne Gewähr für etwaige Beschädigungen.

Das Superangebot:

NEC PC-8000

Computersystem komplett mit Keyboard, Monitor, Doppeldiskettenlaufwerk sowie einem Text- und einem Kalkulationsprogramm.

Komplettpreis: **5 699,—** inkl. MWST

V. Linde elektronic · Neue Straße 18 · 7170 Schwäb. Hall · Tel. 07 91/73 18

Und hier bekommen Sie: Händler gesucht

Proficomputer mit zwei CPU's und **64 K RAM + 12 K ROM** **CPU 6502 + Z80 A** auf dem Motherboard, Int.-Basic + Monitor prog., 7 Slots, alle IC's gesockelt, betriebsbereit Farbe / Grafik apple-compatibel + CP/M-fähig **DM 1320,–**

Light-Pen: High-Resolution, Interface + Software nur **DM 519,—**

Graphic-Table: Software + Hardware, Plott-II kompl. **DM 269,—**

Slimline Disc-Laufwerk 5 1/4 Zoll, FDD-810 ein Qualitätslaufwerk von TEAC. Im Gehäuse, voll apple-compatibel . . . . **DM 688,—**

Disc-Controller-Card . . . . . **DM 159,—**

Disketten höchster Qualität, 1S/125K-40Tr. 10 Stk. nur **DM 49,50**

RGB-Monitor 12", Color, 18 MHz, unbegrenzte Farben **DM 950,—**

Color-Monitor 14", NTSC-Eingang, für Apple-Computer . . . . . nur **DM 690,—**

Monitor 9" grün, High-Resolution 20 MHz, bei uns nur **DM 279,—** IBM-compatible in Kürze lieferbar.

Weiteres COMPUTER-Zubehör finden Sie in unserer SONDERLISTE, kostenlos anfordern!!!

Preise incl. MwSt. + Versand. Apple ist eingetr. Warenz. der Apple Inc.

HÖSCH Elektronic Bruchstr. 43 4000 Düsseldorf 1 Tel. 0211/676214

68008 ECB CPM/68K

- Single-Euro-Karte mit ECB-Bus
- 2 RS232 Vollduplex-Schnittstellen
- 2 Bytewide Sockel für max. 64 kB
- Power fail detect
- 16 kB Monitor: stand-alone lauffähig
- Wartungsfeld kann angeschlossen werden
- Wait synchronisation
- Bustiming: MREQ einstellbar 300 ns – 1 µs
- Interr-Level: NMI, INT, Real-time-clk
- Adressen A0 – A19 ohne Banking
- Refresh-Generierung (ohne Adressen)
- Single 5V-Versorgung
- gepuffertes Uhrenchip
- Baugruppe: DM 1.699 inkl. MWST
- kompl. Rechner: DM 14.799 inkl. MWST
- CPM/68 K für IHREN FD-Controller a.A.

dr.neuhaus

Hochallee 39 2000 Hamburg 13
Tel.: 040 / 44 17 02 Tx: 217 35 13 drn d

Assemblerkurs ASEM-4

komplett nur **DM 98,–**

Mehr rausholen aus dem „Micro" mit schnelleren und leistungsfähigeren Maschinen-Programmen. – Lernen Sie Assembler-Programmierung und Computer-Aufbau verstehen. In direktem Bildschirm-Dialog und begleitet von zahlreichen Übungsaufgaben, wird der Stoff in leichtverständlicher Form vorgestellt.

Im Preis von DM 98,– enthalten ist der **komplette Kurs,** bestehend aus:

- **2 deutschen Handbüchern (ca. 400 Seiten)**
- **Lösungsblättern und Referenzkarte**
- **Software auf Diskette oder Cassette**

Wählen Sie für Ihr System passend aus:

CASSETTE
CBM 30xx
CBM 40xx
VC-20, C-64
ZX-81, ZX-Spectrum
Genie I, II, III
TRS-80, TI-99

DISKETTE
CBM 30xx / 4040
CBM 40xx / 4040
CBM 80xx / 8050
VC-20, C-64 / VC 1541
Genie I, II, III
APPLE II + IIe

. . . andere Systeme auf Anfrage!

Neu: ASEM-4 jetzt auch für CP/M-Systeme

Software für CP/M-Systeme auf Cassette. Die Übertragung auf Ihren Computer erfolgt über die serielle Schnittstelle (V 24) mit einem **normalen Cassetten-Recorder** und dem **mitgelieferten Cassetten-Interface.**
ASEM-4 komplett für CP/M, incl. Cassetten-Interface: nur **DM 148,–**

Versand per NN., Preise incl. MwSt.

Ing.-Büro Wilke

Postfach 17 27 · 5100 Aachen · Telefon 02 41 / 3 06 81 + 87 02 08

LECH-TECHNICS

Gesellschaft zur Herstellung und Vertrieb von elektrischen Geräten und Microcomputern mbH

Nußbaumallee 6
5014 Kerpen-Türnich
West-Germany

Telefon: 0 22 37/81 71
Telex: 889 103 wer d

MICROCOMPUTER PC 301
voll IBM PC kompatibel

Das System enthält:
Mainboard
Disk Driver Card, Multifunktion Card
Color Display Card, Schaltnetzteil
CPU 8088 (4.77 Mhz), DMA Contr. 8237
Interrupt Controller 8259 A
Sockel für Mathematic Chip 8087
128K RAM, aufrüstbar bis 512K
Uhr mit Kalender (Batterie gep.)
zwei Diskettenlaufwerke a' 320K, (doublesided, doubledensity)
1 seriellen Port (RS 232)
1 Parallel Port (Centronic)
Text Mode:
40 Zeichen × 25 Zeilen Farbe/sw
80 Zeichen × 25 Zeilen Farbe/sw
Graphik Mode:
320 Punkte × 200 Linien Farbe/sw
640 Punkte × 200 Linien Farbe/sw
Betriebssystem:
M DOS 1.1 mit ext. BASIC, M DOS 2.0
CP/M 86, ausführliche Handbücher

EINFÜHRUNGSPREIS: **7 998,– DM**

Zubehör für APPLE II und kompatible:

Disk Drive Shugart SA 390	**598,– DM**
Disk Drive Slim Type	**598,– DM**
Monitor "Prince" 24 MHz grün	**395,– DM**
Monitor "Prince" 24 MHz bern.	**445,– DM**
Disk Controller APPLE komp.	**148,– DM**
80 Zeichen Karte softswitch	**198,– DM**
Z 80 Karte (Microsoft)	**148,– DM**
Epromer Karte (2716, 2732)	**128,– DM**
PAL-Karte mit Modulator	**178,– DM**
16K Language Card	**148,– DM**
256K Karte (Pseudo Disk)	**998,– DM**
Graphik Karte für Drucker	**148,– DM**

Paketangebote:

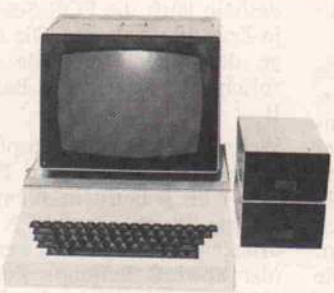
"Atlas 1", 16K Card, Disk Controller, Disk Drive	**1900,– DM**
"Atlas 1", 16K Card, Disk Controller, Disk Drive, Monitor grün	**2300,– DM**
"Atlas 1",16K Card, Disk Controller, 2 Diskettenlaufwerke	**2500,– DM**
"Atlas 1", 16K Card, Disk Controller, Disk Drive, 80 Zeichen Karte, Z 80 Karte, Monitor grün	**2650,– DM**

Alle Preise incl. 14% MwSt. zzgl. Porto und Verpackung.
Ausführliche Informationen und Preislisten gegen DIN A 5 Freiumschlag und 1,30 DM Rückporto.

Microcomputer "Atlas 2" (U 2200), voll APPLE kompatibel

CPU 6502 und Z 80 A, 64 K RAM, 24 K ROM (softswitch), 4 APPLE II kompatible Slots, eingebaute 16K Card, eingebaute Z 80 Card, eingebauter Disk Controller (DOS 3.3) für 2 Diskettenlaufwerke, Schaltnetzteil +5V/5A, +12V/2.5A, -5V/0.5A, -12V/0.5A, abgesetzte Tastatur mit 10er Block, Belegung und Funktion voll APPLE II kompatibel.

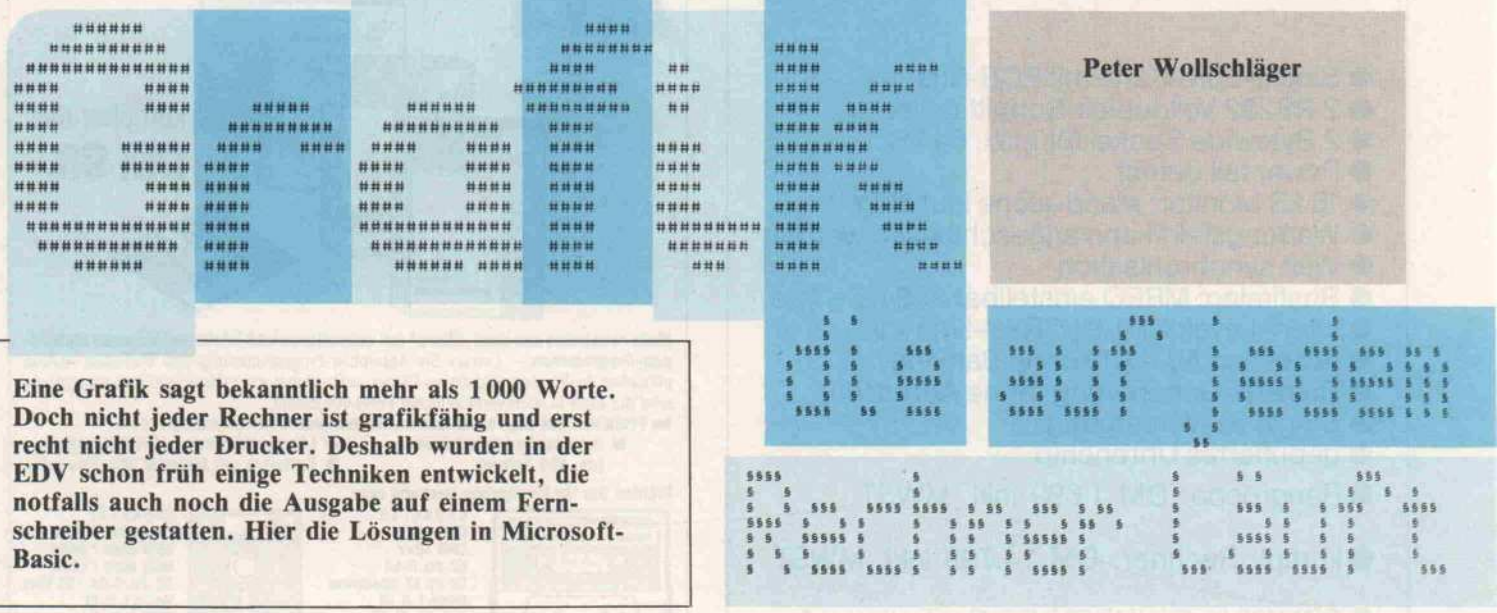
EINFÜHRUNGSPREIS: **1 698,– DM**



Microcomputer "Atlas 1"
voll APPLE kompatibel

betriebsbereit im Aluminiumgehäuse mit UHF-Modulator, 48K RAM, Schaltnetzteil, Anschlüße für Monitor, Cassettenrecorder und Fernseher, Groß/Kleinschreibung.

EINFÜHRUNGSPREIS: **1 098,– DM**



Peter Wollschläger

Eine Grafik sagt bekanntlich mehr als 1000 Worte. Doch nicht jeder Rechner ist grafikfähig und erst recht nicht jeder Drucker. Deshalb wurden in der EDV schon früh einige Techniken entwickelt, die notfalls auch noch die Ausgabe auf einem Fernschreiber gestatten. Hier die Lösungen in Microsoft-Basic.

Am häufigsten findet man Histogramme (Balkendiagramme) in allen möglichen Varianten. Das Prinzip ist — wie immer — ganz einfach. Der Teufel steckt — auch wie immer — im Detail und in einigen Tricks, die sich Programmierer im Laufe der Zeit haben einfallen lassen. Arbeiten wir uns langsam vor.

Bild 1 zeigt ein Listing mit zugehörigem Ergebnis. Die Daten werden in einem Array gesammelt, unabhängig davon, ob sie von der Tastatur, einem externen Gerät oder aus einer Berechnung stammen. Wir werden noch sehen, warum Arrays so vorteilhaft sind. In diesem Beispiel lesen wir die Daten aus der DATA-Zeile in den Array A ein. Damit repräsentiert A(1) Balken 1 mit einer Länge von 2, A(4) Balken 4 (Länge = 7) usw.

Jeder Balken soll in einer Zeile gedruckt werden. Wir haben 8 Daten = 8 Balken, deshalb die FOR-Schleife 1 TO 8 in Zeile 30. Die innere Schleife (Zeile 40) druckt nun bis 'TO A(I)' so viele 'X', wie das I-te Element von A zuläßt. Zeile 50 schließlich hebt den mit Semikolon unterdrückten Zeilenvorschub wieder auf, und mit 'NEXT I' wiederholt sich das Ganze achtmal.

Im allgemeinen muß man die Balken mit Informationen versehen. Wir beschränken uns hier auf die laufende Nummer und den Wert. Wie Sie in Bild 2 sehen, erledigt das Zeile 35 recht gut. In Zeile 55 ist noch ein PRINT hinzugekommen, das eigentlich eine Leerzeile zwischen den Balken bringen sollte. Es zeigt hier keine Wirkung, und das liegt am Drucker. Sollten Sie auch über so ein Exemplar verfügen, müssen Sie da den Vorschub erzwingen, in dem Sie einen String des Inhalts von 81 Blanks ausgeben, wenn die Zeilenlänge 80 ist.

Balken senkrecht

Will man einen Vorgang über der Zeit darstellen, soll beispielsweise jeder Balken einem Monat entsprechen, so ist unsere bisherige Methode unschön, weil der Mensch es gewohnt ist, die Zeit auf der X-Achse zu sehen. Wir müssen also unser Histogramm um 90° drehen (Bild 3). Wir unterstellen, daß Drucker und Bildschirm nur Zeilenvorschub können, was sehr oft die Praxis ist. Jetzt wird die Anzahl der Zeilen durch den Größtwert von A(I), der Länge des längsten Balkens, bestimmt. Wie man den ermittelt, wird später noch gezeigt; wir notieren erst einmal die Zeile 30 G = 10 und gleich daneben die Anzahl Balken mit B = 8. A(G) muß zuerst gedruckt werden, die Zeile(n) mit dem kleinsten Wert zuletzt, deshalb läuft die FOR-Schleife in Zeile 40 rückwärts. Die Länge der inneren Schleife entspricht der Anzahl der Balken B.

Wenn jetzt der Druckkopf innerhalb einer Zeile von Position 1 bis B läuft, muß entweder ein Blank oder ein X gedruckt werden. Blank, wenn I (der abwärts laufende Zeilenzähler) größer ist als A(J),

```
 XX
 XXX             10 DATA 2,3,5,7,10,8,6,4
 XXXXX           20 FOR I=1 TO 8:READ A(I):NEXT
 XXXXXXX         21 '
 XXXXXXXXXX      30 FOR I=1 TO 8
 XXXXXXXX        40    FOR J=1 TO A(I):PRINT "X";:NEXT
 XXXXXX          50    PRINT
 XXXX            60 NEXT I
Ready
```

Bild 1. Die einfachste Art, ein Balkendiagramm zu erzeugen.

```
 1    2  XX            10 DATA 2,3,5,7,10,8,6,4
 2    3  XXX           20 FOR I=1 TO 8:READ A(I):NEXT
 3    5  XXXXX         21 '
 4    7  XXXXXXX       30 FOR I=1 TO 8
 5   10  XXXXXXXXXX    35 PRINT I;TAB(5);A(I);TAB(9);
 6    8  XXXXXXXX      40    FOR J=1 TO A(I):PRINT "X";:NEXT
 7    6  XXXXXX        50    PRINT
 8    4  XXXX          55 PRINT
Ready                 60 NEXT I
```

Bild 2. Der nächste Schritt. Das Histogramm wird beschriftet.

```
          X          10  DATA 2,3,5,7,10,8,6,4
          X          20  FOR I=1 TO 8:READ A(I):NEXT
          X X        21
        X X X        30  G=10:B=8
        X X X X      40  FOR I=G TO 1 STEP -1
      X X X X X      50    FOR J=1 TO B
      X X X X X X 60         IF I>A(J) THEN PRINT "  ";:GOTO 80
  X X X X X X X 70         PRINT"X ":
X X X X X X X X 80     NEXT J
X X X X X X X X 90     PRINT
Ready              100 NEXT I
```

Bild 3. So wird das Histogramm um 90° gedreht.

```
 10  DATA 2,3,5,7,10,8,6,4
 20  FOR I=1 TO 8:READ A(I):NEXT
 21
 30  G=10:B=8
 40  FOR I=G TO 1 STEP -1
 50    FOR J=1 TO B
 60       IF I>A(J) THEN PRINT "  "; ELSE PRINT "X ";
 80    NEXT J
 90    PRINT
100 NEXT I
```

Bild 3a. Mit IF...THEN...ELSE läßt sich das Problem einfacher schreiben.

sprich, die Balkenlänge in dieser Spalte. Die Zeilen 60 und 70 zeigen das, nur ist in beiden Zeilen noch je ein Blank angehängt, um die Abstände zwischen den Balken zu erzeugen. Bild 3a ist nur eine Variante für die Glücklichen, deren Basic IF...THEN...ELSE kennt.

In der Praxis ist es so, daß man zwar die Anzahl Balken weiß, z. B. 12 für 12 Monate, nicht aber ihre Länge. In der Lösung nach Bild 1 kann das dazu führen, daß Balken über den rechten Rand hinausschießen. In der zweiten Lösung besteht die Gefahr zwar nicht, aber wenn der Drucker plötzlich 10 Meter Papier auswirft, ist das ja auch nicht schön.

Um das zu verhindern, wird normiert (Bild 4). Die Zahlen in der DATA-Zeile sind etwas größer geworden, wir wollen aber unseren Ausdruck auf maximal 13 Zeilen begrenzen. Der größte Wert ist 100, daraus muß 13 werden. 13/100 ergibt 0.13. Wenn wir also alle Werte mit dem Faktor 0.13 multiplizieren, ist das Ziel erreicht. In den Zeilen 21/22 finden Sie die Standardroutine zum Suchen eines Größtwertes in einem Array, in 23 den schon erwähnten Faktor. Sollten Sie über einen HP- oder Tek-Rechner verfügen, reicht für alle 3 Zeilen 'F = 13/MAX(A)'.

In Zeile 24 wird der Array mit F multipliziert, und dann folgt alles wie gehabt, bis auf Zeile 45, die jetzt den neuen Skalenwert ausgibt. Die Zahlen sind evtl. etwas nichtssagend, weshalb üblicherweise der Array vorab in Prozent umgerechnet wird. Dann kann man entweder akzeptieren, daß die Länge des Ausdrucks vom größten Prozentsatz abhängt, oder man normiert den auch noch.

Der nächste Schritt: Oft sollen Daten-Paare, zum Beispiel Umsatz und Erlös, zusammen dargestellt werden. Das eine als Reihe von 'X', das andere als '*'. Wie Bild 5 zeigt, muß dazu nur die innere Schleife zweimal gefahren werden, 'no problem'.

Schwieriger wird es schon, wenn wir das Bild drehen wollen. Dann gibt es nämlich je Druckposition 4 Möglichkeiten, und zwar (B steht für Blank): BB, BX, *B und *X. Bild 6 zeigt die Lösung. Im Prinzip doch ganz einfach: Die 4 Möglichkeiten werden mit 3 'IF...THEN' getestet, die vierte ergibt sich dann zwangsläufig.

```
100                     X
92.3077                 X
84.6154                 X
76.9231                 X X
69.2308               X X X
61.5385               X X X
53.8462               X X X X
46.1538             X X X X X
38.4615             X X X X X X
30.7692             X X X X X X
23.0769           X X X X X X X
15.3846         X X X X X X X X
7.69231         X X X X X X X X
Ready
```

```
10 DATA 20,30,50,70,100,80,60,40
20  FOR I=1 TO 8:READ A(I):NEXT
21 G=A(1):B=8
22 FOR I=2 TO 8:IF A(I)>G THEN G=A(I):NEXT
23 F=13/G
24 FOR I=1 TO 8:A(I)=A(I)*F:NEXT
25 '
40  FOR I=13 TO 1 STEP -1
45  PRINT I/F,
50    FOR J=1 TO 8
60       IF I>A(J) THEN PRINT "  ";:GOTO 80
70       PRINT"X ";
80    NEXT J
90    PRINT
100 NEXT I
```

Bild 4. Durch Normierung erreicht man konstant große Ausdrucke.

Kurven angedeutet

Warum immer nur Balken? Kurven kann man zwar nicht

```
XXX
*****
XXXXXXX
********
XXXXXXXXXXX
**********
XXXXXXXXXXXXX
**********
Ready
```

```
10 DATA 3,7,11,14
11 DATA 5,8,12,11
20 FOR I=1 TO 4:READ A(I):NEXT
21 FOR I=1 TO 4:READ V(I):NEXT
30 FOR I=1 TO 4
40     FOR J=1 TO A(I):PRINT "X";:NEXT
41     PRINT
42     FOR J=1 TO V(I):PRINT "*";:NEXT
50     PRINT:PRINT
60 NEXT I
```

Bild 5. Daten-Paare zugleich dargestellt.

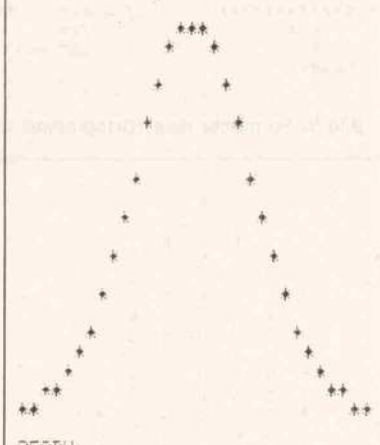
```
                *   X
                *   X
              X *   *X
            *X *X *X
          X *X *X *X *
        *X *X *X *X *X
        *X *X *X *X *X *
  X *   *X *X *X *X *X *
*X *X *X *X *X *X *X *X
*X *X *X *X *X *X *X *X
Ready
```

```
10 DATA 2,3,5,7,10,8,6,4
11 DATA 3,2,6,8,7,10,5,2
20 FOR I=1 TO 8:READ A(I):NEXT
21 FOR I=1 TO 8:READ V(I):NEXT
30  G=10:B=8
40  FOR I=G TO 1 STEP -1
50    FOR J=1 TO B
60       IF I>A(J)  AND I>V(J)  THEN PRINT "   ";:GOTO 80
70       IF I>A(J)  AND I<=V(J) THEN PRINT " X ";:GOTO 80
71       IF I<=A(J) AND I>V(J)  THEN PRINT "*  ";:GOTO 80
72       PRINT "*X ";
80    NEXT J
90    PRINT
100 NEXT I
```

Bild 6. Bild 5 um 90° gedreht und mit Zwischenräumen: sieht schon besser aus.

```
1 SYSTEM "ROUTE *DO *PR"
100 SI=1:MY=5:H=8:PI=3.14
110 K1=H/SI*SQR(2*PI)
120 DEF FN GS(X)=INT(K1*EXP(-((X-MY)/SI)^2/2)+.5)
130 FOR I=MY-3*SI TO MY+3*SI STEP .5
140 PRINT TAB(FN GS(I));"*"
150 NEXT
160 FOR I=FN GS(MY) TO 0 STEP -1
170 FOR J=MY-3*SI TO MY+3*SI STEP .2
180 IF FN GS(J)=I THEN PRINT "*";:GOTO 200
190 PRINT" ";
200 NEXT J
210 PRINT
220 NEXT I
221 SYSTEM "RESET *DO"
```

Bild 7. So druckt man Kurven, der Drucker verzerrt das Bild.



READY.

Bild 8. Das Programm aus Bild 7, aber mit einem anderen Drucker.

durchziehen, aber immerhin ganz gut andeuten. In Bild 7 sind in den Zeilen 100—110 ein paar Konstanten definiert, die Funktion in Zeile 120 ist die der Gauß'schen Glockenkurve, die sog. Normalverteilung. Wir kommen noch drauf, erst einmal zum Drucken von Funktionen überhaupt. Die Kurve von links nach rechts abzubilden, ist höchst simpel, die Zeilen 130—150 erledigen das. Als TAB-Parameter wird einfach der Funktionswert eingesetzt.

Die zweite Kurve ist schon schwieriger. Vorab: Wenn Sie da die Glockenform kaum erkennen, so liegt das nicht am Programm, sondern am Drucker. Dieser hat (solange man ihn nicht besonders einstellt) ein unsymmetrisches Zeilen/Spaltenverhältnis, was leider unsere Kurve linearisiert. Bild 8 zeigt das gleiche Programmergebnis auf einem Commodore-Drucker.

Wie müssen wieder vom Größtwert rückwärts drucken. Die Maxima-Suche können wir hier sparen, das ist bei dieser Funktion GS(MY), so Zeile 160. Die innere Schleife muß in ziemlich kleinen Schritten laufen (hier Step 0.2), sonst haben wir nämlich geringe Chancen, einen Funktionswert zu erwischen, der gleich I, also unserer Zeilennummer, ist. Die Zeilen 1 und 221 können Sie ignorieren, sie dienen lediglich bei dem hier eingesetzten Rechner zur Umleitung der Bildschirmausgabe auf den Drucker.

Bliebe last not least, wie druckt man Histogramm und Kurve gleichzeitig? In Forschung und Technik tägliche Praxis ist das Auswerten von Meßreihen. Dazu werden die Meßwerte in Klassen sortiert (gezählt). Die Klassenhäufigkeit kann man als Balkendiagramm darstellen. Man kann aber aus den Meßwerten die Verteilungsfunktion errechnen. Letzteres setzt immer voraus, daß ein Modell, wie z. B. die Gauß-Verteilung, angenommen wird. Stimmt die Annahme nicht, sind alle errechneten Werte wie etwa Mittelwert und Streuung falsch. Es gibt zwar zahllose Programme, die automatisch die Ist-Verteilung gegen verschiedene Modelle rechnen und Unmengen von Zahlen ausdrucken, aber wer will das alles lesen?

Mehr als alle Zahlen sagt eine Grafik, die die Ist-Verteilung ausdruckt und darüber als Kurve das Modell, von dem das Programm meint, daß es das am besten korrelierende sei. Mit einem Plotter ist das einfach, da zeichnet man Histogramm und Kurve nacheinander. Hat man nur einen Drucker, muß man sich an ein Programm wie das in Bild 9 halten. Wie man die Parameter der Kurve rechnet, wird nicht vorgeführt. Die DATA-Zeile entspricht schon der Klassenhäufigkeit, in Zeile 40 stehen schon die Rechenergebnisse in Konstanten, Zeile 50 rechnet den Array in Prozent um, und in 60 schließlich finden Sie wieder die Funktion der Gauß-Laplace-Verteilung. Wir wollen nun den Array-Inhalt als Balken mit 'X' drucken und die Kurve mit '*'. Dabei hat ein '*' Vorrang, das heißt schneidet die Kurve einen Balken, soll kein 'X' sondern '*' gedruckt werden.

Letzterer Fall bedeutet, es müssen erst einige 'X', dann ein '*' und schließlich die restlichen 'X' gedruckt werden. Die Zeilen 140 bis 170 erledigen das. Es kann aber auch sein, daß die Kurve über den Balken liegt, dann sind erst die 'X', dann einige Blanks und dann der '*' zu drucken. Diesen Teil erledigen die Zeilen 190—210.

Der Trick in dem Programm ist, daß die Endwerte der FOR-Schleifen vom Index abhängige Variable sind. In Zeile 190 steht z. B. 'FOR J=1 TO T+(T=A(I))'. '(T=A(I))' ist ein logischer Ausdruck, der als wahr —1 ergibt. (So alle Commodore- und Tandy-Rechner. Beim Apple ergibt sich +1, da sind dann die Vorzeichen zu ändern.) Diese Technik erspart diverse IF-THEN zu verschiedenen Schleifen, die sich nur in der Endbedingung unterscheiden.

In Bild 10 wird die Gauß-Verteilung wieder aufgerichtet. (Die Daten sind da andere.) Wie Sie sehen, ist die Lösung wesentlich einfacher als die aus Bild 9. Daß man dennoch so oft die 'falsche' Darstellung findet, liegt daran, daß die wesentlich schneller ist. Der Nachteil des Programms von Bild 10 ist, daß die Funktion in der inneren Schleife gerechnet wird, also sehr oft, und das kostet Zeit.

Zum Schluß: Es gibt Programme, die entsprechend dem späteren Druckbild eine Maske generieren. Die würden also, um bei unserem letzten Beispiel zu bleiben, einen 2-dimensionalen Array mit Blanks, Kreuzen und Sternchen laden. Die Technik ist sehr einfach, da man ja wie beim Plotter alle Vorgänge nacheinander erledigen kann. Sie empfiehlt sich, wenn man weitere Balken mit anderem Muster und/oder mehr als eine Kurve ins Bild bringen will. Wie auch immer, diese Art der Grafik ist zwar nicht hochauflösend, aber höchst universell. Das gilt für die Hardware ebenso wie die Software. In den Grafik-Befehlen unterscheiden sich die Rechner sehr stark, PRINT können sie alle. □

```
*
X*XX                    10 DIM A(11)
  *                     20 DATA 1,2,3,5,3,3,3,1,2,3
XXX*XXXX                30 FOR I=1 TO 10:READ A(I):NEXT
   *                    40 N=25:M=5.2:S=2.6:K=10:W=.95:U=16.2
XXXXX*XXXXXX            50 FOR I=1 TO 10:A(I)=INT((100*A(I)/N)/W):NEXT
     *                  60 DEF FN GS(X)=INT(U*EXP(-((X-M)/S)^2/2))
XXXXXXXXX*XXXXXXXXXXX   70 '-----------------------------------------
          *             100 FOR I=1 TO K+1
XXXXXXXXXXX *           110    PRINT TAB(FN GS(I-2)+1)"*"
             *          120    T=FN GS(I-1.5)
XXXXXXXXXXX  *          130 '
              *         140    IF A(I)>T THEN 190
XXXXXXXXXXX   *         150       FOR J=1 TO A(I):PRINT"X";:NEXT
             *          160       FOR J=A(I) TO T-2-(A(I)<>0):PRINT" ";:NEXT:PRINT"*"
XXXX          *         170       GOTO 230
            *           180 '
XXXXXXX  *              190    FOR J=1 TO T+(T=A(I)):PRINT"X";:NEXT:PRINT"*";
          *             200    IF T<>A(I) THEN  FOR J=1 TO A(I)-T-1:PRINT"X";:NEXT
XXXXXX*XXXX             210    PRINT
    *                   220 '
   *                    230 NEXT I
Ready
```

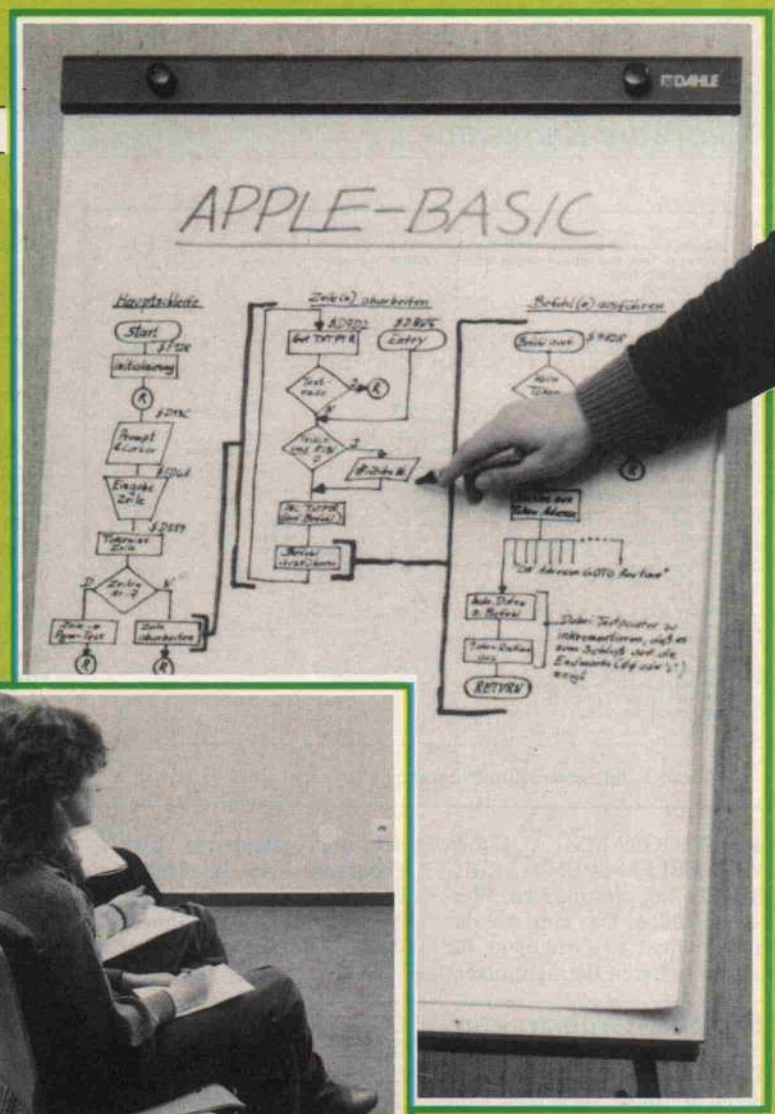
Bild 9. So mischt man Histogramme und Kurven.

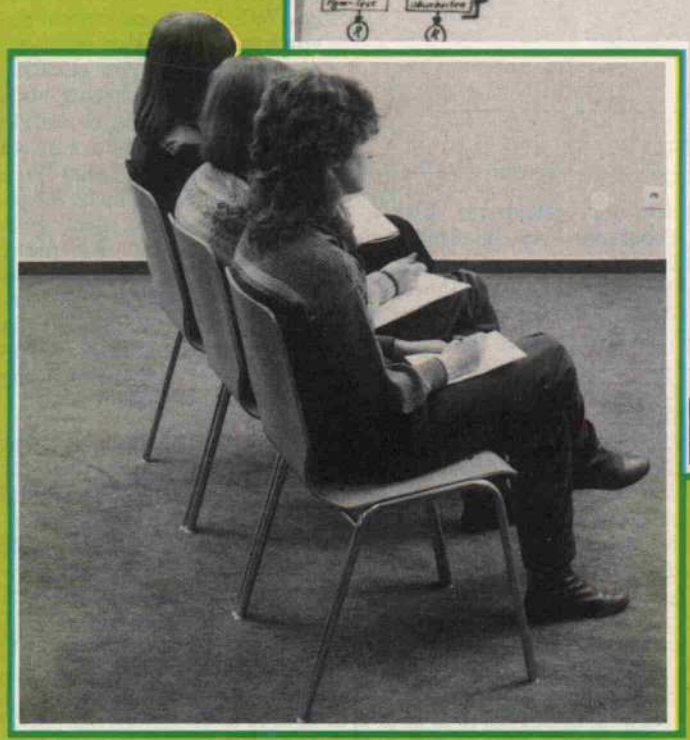
```
                                1 OPEN 4,4 CMD4
          *  X                  10 DIM A(12)
          X  X                  20 DATA 1,2,3,3,5,4,3,2,1,1
       *  X  *                  30 FOR I=1 TO 10:READ A(I):NEXT
          X  X                  40 N=25:M=5:S=2   K=10:W=.9:U=16
    X  X  X  X  X               50 FOR I=1 TO 10:A(I)=INT((100*A(I)/N)/W):NEXT
    X  X  X  X  X               60 G=16
    *  X  X  X  *               70 DEF FN GS(X)=INT(U*EXP(-((X-M)/S)↑2/2)+.5)
    X  X  X  X  X               80 REM ---------------------------------------
 X  X  X  X  X  X  X            90 FOR I=G TO 1 STEP-1
 X  X  X  X  X  X  X           100 FOR J=1 TO K+1
 *  X  X  X  X  X  *           110 IF FNGS(J)=I THEN PRINT"*  ";:J=J+1
 X  X  X  X  X  X  X           120 IF A(J)>I THEN PRINT "X  ";:GOTO 140
 X  X  X  X  X  X  X  X  X     130 PRINT"   ";
 *  X  X  X  X  X  X  X  *     140 NEXT J
 X  X  X  X  X  X  X  X  *     150 PRINT
                               160 NEXT I
READY.                        READY.
```

Bild 10. Das Histogramm gedreht. Besser, kostet aber mehr CPU-Zeit.

BASIC intern Teil 1

Was nicht im Handbuch steht





Heinz-Peter Heidinger

Wie funktionieren Interpreter und Compiler? Wie kann man das 'serienmäßige' BASIC erweitern? Mit welchen Tricks arbeiten die Programmierer von Microsoft? — Eine c't-Serie für alle, die mehr als die 'Benutzeroberfläche' ihres Computers kennenlernen wollen. Also auch für Assembler-Freaks.

BASIC zu erlernen, stellt meist kein großes Problem dar. Manche Dinge sind allerdings nicht auf Anhieb verständlich. Als Beispiel sei hier nur VARPTR-Funktion genannt. Diese Funktion liefert Ergebnisse, die die Art und Weise reflektieren, wie BASIC sich Variablennamen und -inhalte intern im Speicher ablegt. Ohne spezielle Kenntnisse über die Strukturen der Speicherbereiche, die BASIC intern benutzt, fällt es schwer, mit diesen Werten etwas anzufangen. Dieser Artikel liefert das nötige Know-how.

Zum Verständnis sollten Sie lediglich etwas Übung im Umgang mit binären und hexadezimalen Zahlen mitbringen. Es wäre auch von Vorteil, wenn Ihnen ein Monitor-Programm zur Verfügung stände. Ein solches ist zum Beispiel in c't 12/83 für Z80-Systeme vorgestellt worden. Wer keinen Monitor zur Verfügung hat, wird die folgenden Strukturuntersuchungen an BASIC 'trocken' nachvollziehen können. Die Untersuchungen im Rahmen dieses Artikels orientieren sich am BASIC-Interpreter des TRS-80/Modell-1, sind aber prinzipiell auf die weiteren Interpreter aus dem Hause Microsoft übertragbar.

Bevor wir auf die ersten grundsätzlichen Strukturen von BASIC zu sprechen kommen, einige Worte zum SETUP-Programm in Tabelle 1. Dieses Programm dient einerseits dazu, für unsere Untersuchungen einen PROGRAMMTEXT im PROGRAMM-SPEICHER-BEREICH abzulegen. Es enthält alle möglichen Datenstrukturen (Datentypen), die in einem realen Programm in Erscheinung treten könnten — ganzzahlige, einfach- und doppeltgenaue sowie Zeichenketten-Variablen und Konstanten, benutzerdefinierte Funktionen, BASIC-Anweisungen, -Funktionen und -Befehle. Andererseits hat es die Aufgabe, durch

```
----------------------------------------------------
↓     Programmtext
----------------------------------------------------
↓     einfache Variable
----------------------------------------------------
↓     Feldvariable
----------------------------------------------------
      freier Speicher
----------------------------------------------------
↑     Stapel (Stack)
----------------------------------------------------
↑     Zeichenkettenvariable
          (String space)
----------------------------------------------------
      fuer Maschinensprache-Programme
      reservierter Speicherplatz sofern
      dieser durch MEM SIZE geschuetzt
      wurde
----------------------------------------------------
```

Bild 1. Grobstruktur des Speichers für BASIC

```
100 REM *** Programm-Anfang ***
110 REM *** Speicher-Setup fuer: "BASIC intern!"
120 CLEAR 1000
130 DEFINT A
140 DEFSNG C
150 DEFDBL E
160 DEFSTR G
170 DEF FN I(A,B) = A(B
180 DIM A1(7), C1(7), E1(7), G1(7)
190 DIM A2(7,7)
200 DIM A3(7,7,7)
210 A=32000:                    BX=20000
220 C=45678.9:                  D!=56789.1
230 E=123456789.1234567:        F#=234567890.2345678
240 G="String - vordefiniert": H$="String - explizit definiert"
250 INPUT"ganzzahliger    Wert: ";AI
260 INPUT"einfach genauer Wert: ";CI
270 INPUT"doppelt genauer Wert: ";EI
280 INPUT"String              : ";GI
290 FOR X=0 TO 7: A1(X) = A       + X          :NEXT
300 FOR X=0 TO 7: C1(X) = C       + X + .1     :NEXT
310 FOR X=0 TO 7: E1(X) = E       + X + .000001 :NEXT
320 FOR X=0 TO 7: G1(X) = "Feld-" + G + STR$(X) :NEXT
330 FOR X=0 TO 7
340 FOR Y=0 TO 7
350 A2(X,Y) = 2(X + 2(Y
360 NEXT: NEXT
370 FOR X=0 TO 7
380 FOR Y=0 TO 7
390 FOR Z=0 TO 7
400 A3(X,Y,Z) = 2(X + 2(Y +2(Z
410 NEXT: NEXT: NEXT
420 PRINT FN I(5,2)
```

Tabelle 1. Speicher-Setup-Programm

den PROGRAMMLAUF den VARIABLEN-SPEICHERBEREICH mit bestimmten Werten zu füllen. Das eine wie das andere dient als Grundlage für unsere späteren Betrachtungen.

Speicher-Grobstruktur

Wenn wir einmal die Speicherbereiche nicht beachten, die durch den BASIC-Interpreter selbst, den Bildwiederholspeicher und sonstige rechnerabhängige Gegebenheiten belegt werden, gliedert sich der für BASIC verfügbare Speicherbereich schematisch, wie in Bild 1 dargestellt. Die Pfeile geben dabei jeweils an, in welcher Richtung sich die Belegung der einzelnen Speichersegmente beim Schreiben des Programmes beziehungsweise beim Programmlauf ändert.

Die Unterscheidung zwischen 'Schreiben (oder Ändern) des Programmes' und 'Programmlauf' ist natürlich nicht willkürlich. Wie aus Bild 1 leicht zu folgern ist, 'schrumpft' der Speicherabschnitt 'Programmtext' in Richtung niedrigerer Speicheradressen, wenn Sie Zeilen löschen, oder er breitet sich in Richtung höherer Speicheradressen aus, wenn Sie Programmzeilen hinzufügen, und überschreibt damit die Speicherbereiche, in denen die Variablenwerte abgelegt sind, wenn das Programm schon einmal gelaufen ist. Dagegen ist dieser Speicherabschnitt statisch, wenn das Programm läuft.

Vielleicht haben Sie schon einmal versucht, ein abgebrochenes und geändertes BASIC-Programm anschließend mit CONT wieder fortzusetzen. In üblicher Manier hat Ihnen der Rechner darauf dann sicher zu verstehen gegeben, daß er dazu nicht in der Lage ist — CAN'T CONTINUE! Nun, Ihr spitzfindiger mikroelektronischer 'Zeitgenosse' hat gemerkt, daß Sie etwas an den Speichergrenzen des Programmspeicherbereiches geändert haben und war folglich nicht mehr in der Lage, den Variablen-Speicherbereich zu finden. Den baut er erst dann wieder auf, wenn Sie das Programm neu starten. In seinem 'Workspace' (Arbeitsbereich des BASIC-Interpreters, in dem sich BASIC alles 'Wichtige' merkt, siehe Tabelle 2), legt sich der Interpreter unter vielen anderen auch die Adressen für Anfang und Ende des Programmtextes ab. Man nennt diese beiden Adressen auch HEAD- und TAIL-Pointer (wörtlich übersetzt: 'Kopf'- und 'Schwanz-Zeiger' und nachfolgend immer als HP und TP abgekürzt). Zum HP erfahren Sie später noch einiges. Wir gehen erst einmal davon aus, daß er sich nicht ändert. Interessant ist zunächst der TP, dem gleichzeitig noch eine weitaus wichtigere Aufgabe zukommt. In Bild 1 sehen wir, daß sich BASIC alles 'Wissenswerte' über einfache Variable direkt hinter dem Programmtext ablegt. Wo dieser Bereich im Speicher beginnt, markiert der TP ebenso. Anders gesagt: der Zeiger auf das Programmtextende ist gleichzeitig Zeiger auf den Anfang der Variablentabelle.

Direkt auf das Speichersegment für einfache Variable folgt der Abschnitt, in dem sich BASIC die Namen und korrespondierenden Werte für Feld-Variable ablegt. Auch für diesen Speicherabschnitt ist im Workspace ein Zeiger vorgesehen: der Feldvariablen-Pointer (FPA). Dieser Zeiger ändert sich immer dann, wenn in die Tabelle für einfache Variable eine Variable neu aufgenommen wird. Dann muß der Feldvariablen-Bereich auch in Richtung des freien Speichers verschoben werden. Damit BASIC auch 'weiß', wie groß der Feldvariablen-Block ist, 'merkt' es sich außerdem, wo dieser Block im Speicher endet. Demzufolge gehören zu diesem Speichersegment zwei Pointer: der Zeiger auf den Anfang (FPA) und der auf das Ende (FPE).

Im weiteren wollen wir die Speicherschematik 'von unten her' (bezogen auf Bild 1) betrachten. Gehen wir einmal davon aus, daß kein Speicherplatz für ML-Routinen (ML = 'machine language') reserviert wurde. Dann befindet sich am oberen Speicherende, das heißt, auf den höchsten Speicheradressen, der 'Stringspace'. Dieser Speicherbereich enthält die Zeichenfolgen, die Stringvariablen zugewiesen wurden. Schaut man mit einem Monitor in diesen Abschnitt, stellt man fest, daß sich hier ohne erkennbare 'Trennbytes' und völlig beziehungslos alle möglichen Zeichenfolgen tummeln. Daß hier eine Beziehung zu der entsprechenden Stringvariablen besteht und Trennzeichen gar nicht notwendig sind, werden wir sehen, wenn wir die Variablentabelle näher betrachten.

Wichtig ist jedoch zu wissen, daß der Stringspace der Speicherbereich ist, dessen Umfang Sie mit der CLEAR-Anweisung festlegen. Damit ist auch dieses Speichersegment, wie das oberste (Programmtext), statisch, was allerdings nicht bedeutet, daß sich sein Inhalt nicht ändert. Ganz im Gegenteil! In diesem Speicherbereich regt sich am meisten, denn hier findet die berüchtigte 'garbage collection' statt. Was das ist, erfahren Sie bei näherer Betrachtung der Variablen-Tabelle.

```
# DUMP  40A0  411A
40A0  16 BC FF FF 46 6A 00 CA 64 01 00 00 00 00 00 02  *....Fj..d.......*
40B0  00 FE BF B5 40 06 29 19 02 57 BE 00 00 00 00 00  *....@.)..W......*
40C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  *................*
40D0  00 00 00 06 29 19 3A BE CC 64 00 00 00 00 00 AA  *....).:..d......*
40E0  6D 00 00 00 00 00 C7 64 14 BC 00 00 00 00 00 00  *m......d........*
40F0  00 00 00 CC 64 A4 01 B9 6D BC 6D 36 6E 74 73 45  *....d...m.m6ntsE*
4100  6A 02 04 04 04 08 04 03 04 04 04 04 04 04 04 04  *j...............*
4110  04 04 04 04 04 04 04 04 04 04 04                 *...........     *
```

1. Anfang des 'string space'
2. Aktuelle Zeilenummer. Wir bei Rückkehr zur Interpreter-hauptschleife auf 65535 (FFFFh) gesetzt! => Direkt-Modus!
3. HEAD-Pointer. Zeigt auf den Anfang des Programmtextes (entspr. Betriebssystem!)
4. Höchste Speicheradresse, die für BASIC verfügbar ist (TOPMEM). Ab dieser Adresse+1 beginnt der für Maschinen-Programme reservierte Speicherbereich sofern dieser durch MEM SIZE festgelgt wurde.
5. Stack-Basis-Adresse (immer 2 Byte über dem String space).
6. TAIL-Pointer / Anfang der Variablen-Tabelle
7. Anfang Feld-Variablen
8. Ende Feld-Variablen
9. DATA-Pointer. Zeigt auf das Trennzeichen hinter dem zuletzt mit READ gelesenen Datum. Bei Programmstart (RUN) wird er auf den Anfang des Programmtextes initialisiert (Zeilen-Trennzeichen '00').
10. Typcode-Tabelle. Enthält für jeden Anfangsbuchstaben der Variablennamen den entsprechenden Typcode. Durch NEW werden alle Tabellenwerte auf 04 gesetzt (Typcode für einfache Genauigkeit SNG). Die Befehle zur Typcodefestlegung (DEFINT, DEFSNG, DEFDBL, DEFSTR) manipulieren die Tabelle. Für Variablen, die keine Typkennung (%, !, #, $) mitführen, wird der Typcode dieser Tabelle entnommen.

Tabelle 2. Microsoft-BASIC's Workspace (Auszug) / hier: TRS-80/M1/L3 (NEWDOS/80)

Prozessor-Stack und BASIC-Stack

Über dem Stringspace ist der BASIC-Stack im Speicher angeordnet. Wie üblich dehnt er sich in Richtung niedrigerer Speicheradressen aus, wenn er zum Abstapeln von Programm-

hex	dez.	ASCII	hex	dez.	ASCII	hex	dez.	ASCII
20	32	Space	40	64	@	60	96	`
21	33	!	41	65	A	61	97	a
22	34	"	42	66	B	62	98	b
23	35	#	43	67	C	63	99	c
24	36	$	44	68	D	64	100	d
25	37	%	45	69	E	65	101	e
26	38	&	46	70	F	66	102	f
27	39	'	47	71	G	67	103	g
28	40	(	48	72	H	68	104	h
29	41	)	49	73	I	69	105	i
2A	42	*	4A	74	J	6A	106	j
2B	43	+	4B	75	K	6B	107	k
2C	44	,	4C	76	L	6C	108	l
2D	45	-	4D	77	M	6D	109	m
2E	46	.	4E	78	N	6E	110	n
2F	47	/	4F	79	O	6F	111	o
30	48	0	50	80	P	70	112	p
31	49	1	51	81	Q	71	113	q
32	50	2	52	82	R	72	114	r
33	51	3	53	83	S	73	115	s
34	52	4	54	84	T	74	116	t
35	53	5	55	85	U	75	117	u
36	54	6	56	86	V	76	118	v
37	55	7	57	87	W	77	119	w
38	56	8	58	88	X	78	120	x
39	57	9	59	89	Y	79	121	y
3A	58	:	5A	90	Z	7A	122	z
3B	59	;	5B	91	[	7B	123	{
3C	60	<	5C	92	\	7C	124	|
3D	61	=	5D	93	]	7D	125	}
3E	62	>	5E	94	^	7E	126	~
3F	63	?	5F	95	_	7F	127	DEL

Tabelle 3. ASCII-Code-Tabelle

daten gebraucht wird. Neben seiner Prozessor-üblichen Funktion dient er dem Interpreter dazu, die Rückkehradressen für Unterprogrammaufrufe abzulegen. 'Das ist doch normal', mögen Assembler-Freaks jetzt gedacht haben. Nun, ganz so 'normal' ist es eben nicht, denn als Rückkehradresse für Unterprogrammaufrufe durch GOSUB, ist der Prozessor-interne 'programm counter' (PC) völlig irrelevant! Der Interpreter muß 'wissen', an welcher Stelle *im Programmtext* der UP-Aufruf stattgefunden hat, und die Speicheradresse dieser Stelle muß daher auf dem Stapel abgelegt sein.

Folgender Satz ist daher fundamental für BASIC-Interpreter auf 8080/8085- und Z 80-Rechnern:

Zeiger auf den Programmtext ist immer das HL-Register. Sie können in Maschinensprache-Routinen mit den Registern machen, was Sie wollen (ausgenommen den Prozessor-Stackpointer zu ändern, ohne ihn wiederherzustellen) — unter der Bedingung, daß das HL-Register bei Rückkehr zum Interpreter unmittelbar hinter die Aufrufstelle zeigt!

Diesem Satz zufolge muß also bei UP-Verzweigungen der Inhalt des HL-Registers (BASIC-PC) auf dem Stapel abgelegt werden.

Auf dem BASIC-Stack werden ferner die Programmtext-Adressen der ersten Anweisung nach Öffnen einer FOR-NEXT-Schleife sowie Name und Wert der korrespondierenden Laufvariablen abgelegt, damit BASIC weiß, ab wo weiter zu interpretieren ist, wenn eine NEXT-Anweisung auftritt.

Zwischen den beiden Speicherbereichen, deren Grenzen sich beim Programmlauf dynamisch verändern (Stack und einfache/Feld-Variablentabelle), liegt als Pufferzone der freiverfügbare Speicherraum ('spare memory'). Wenn dieser aufgezehrt ist (sei es durch zahlreichen Variablengebrauch, viele und/oder groß dimensionierte Felder, vielfältig verschachtelte FOR-NEXT-Schleifen oder auch zahlreiche UP-Aufrufe), erzeugt der BASIC-Interpreter einen 'OUT-OF-MEMORY ERROR'.

Blick ins 'Kleingedruckte'

Nach dem vorangegangenen Überblick wollen wir jetzt die einzelnen Speicherabschnitte detailliert betrachten. Beginnen wir mit dem Abschnitt 'Programmtext'. Einen Auszug dieses Abschnittes sehen Sie in Tabelle 5. Auf den ersten Blick wird man erstaunt sein, keine BASIC-Schlüsselwörter zu finden. Nun, sie sind dennoch vorhanden, aber in verschlüsselter Form.

Bekanntlich wird auf (fast) allen Rechnern zur Darstellung von Zeichen der ASCII-Zeichensatz (American Standard Code for Information Interchange) verwendet. Eine Code-Tabelle dieses Zeichensatzes finden Sie in Bild 2. Diese Codeliste endet bei 127 (7Fh). Mit acht Bit Datenbreite ist es aber möglich, 128 weitere Codes für andere Zwecke zu vergeben.

Exakt dies wird auch mit den BASIC-Schlüsselwörtern gemacht. Daher finden Sie auch keine Keywords mehr im Speicher: Der Interpreter hat selbige in einzelne Bytes übersetzt, die 'Token' genannt werden (token <engl.> = Kennzeichen, Kennzeichner), bevor er die Programmzeile im Speicher ablegte. Eine Token-Liste, wie sie für den TRS-80 gilt, zeigt Tabelle 4.

Das 'Tokisieren' hat entscheidenden Einfluß auf die Arbeitsgeschwindigkeit des Interpreters, denn über den Code des Schlüsselwortes kann er sehr schnell die Adresse der zugehörigen Interpreter-Routine in einer Sprungtabelle finden. Außerdem erübrigt sich die zeitraubende Unterscheidung zwischen Variablennamen und Schlüsselwörtern während des Programmlaufes, denn alles, was dann kein Token ist, wird als Variable interpretiert. Unterschieden wird schon bei der Übersetzung (bevor die Zeile im Programmspeicherbereich abgelegt wird) und nicht erst beim Programmlauf.

Inspizieren wir nun den Aufbau einer Programmzeile: Dem Memory-Dump in Tabelle 5 entsprechend finden wir das im Speicher befindliche Programm gemäß Tabelle 1 ab Speicheradresse 6A46h, wie durch den HP (in Tabelle 2) ausgewiesen ist. Tatsächlich beginnt das BASIC-Programm aber schon ein Byte vorher, und dieses Byte muß immer '00' sein. Zunächst scheinen die ersten sechs Bytes bedeutungslos zu sein. Ab Byte sieben finden wir dann den Text, der in Zeile 100 hinter dem REM-Befehl abgelegt wurde, und an den sich wieder ein '00'-Byte anschließt. Die Funktion dieses '00'-Bytes ist jetzt augenfällig: es dient BASIC als Trennzeichen zwischen zwei Programmzeilen.

Betrachten wir jetzt die nächsten fünf Bytes der Zeile: Die Bytes 2 und 3 lauten 64h und

80 - 128	END	90 - 144	RESTORE	A0 - 160	OUT			
81 - 129	FOR	91 - 145	GOSUB	A1 - 161	ON			
82 - 130	RESET	92 - 146	RETURN	A2 - 162	OPEN			
83 - 131	SET	93 - 147	REM	A3 - 163	FIELD			
84 - 132	CLS	94 - 148	STOP	A4 - 164	GET			
85 - 133	CMD	95 - 149	ELSE	A5 - 165	PUT			
86 - 134	RANDOM	96 - 150	TRON	A6 - 166	CLOSE			
87 - 135	NEXT	97 - 151	TROFF	A7 - 167	LOAD			
88 - 136	DATA	98 - 152	DEFSTR	A8 - 168	MERGE			
89 - 137	INPUT	99 - 153	DEFINT	A9 - 169	NAME			
8A - 138	DIM	9A - 154	DEFSNG	AA - 170	KILL			
8B - 139	READ	9B - 155	DEFDBL	AB - 171	LSET			
8C - 140	LET	9C - 156	LINE	AC - 172	RSET			
8D - 141	GOTO	9D - 157	EDIT	AD - 173	SAVE			
8E - 142	RUN	9E - 158	ERROR	AE - 174	SYSTEM			
8F - 143	IF	9F - 159	RESUME	AF - 175	LPRINT			
B0 - 176	DEF	C0 - 192	VARPTR	D0 - 208	/			
B1 - 177	POKE	C1 - 193	USR	D1 - 209	[			
B2 - 178	PRINT	C2 - 194	ERL	D2 - 210	AND			
B3 - 179	CONT	C3 - 195	ERR	D3 - 211	OR			
B4 - 180	LIST	C4 - 196	STRING$	D4 - 212	>			
B5 - 181	LLIST	C5 - 197	INSTR	D5 - 213	=			
B6 - 182	DELETE	C6 - 198	POINT	D6 - 214	<			
B7 - 183	AUTO	C7 - 199	TIME$	D7 - 215	SGN			
B8 - 184	CLEAR	C8 - 200	MEM	D8 - 216	INT			
B9 - 185	CLOAD	C9 - 201	INKEY$	D9 - 217	ABS			
BA - 186	CSAVE	CA - 202	THEN	DA - 218	FRE			
BB - 187	NEW	CB - 203	NOT	DB - 219	INP			
BC - 188	TAB(	CC - 204	STEP	DC - 220	POS			
BD - 189	TO	CD - 205	+	DD - 221	SQR			
BE - 190	FN	CE - 206	-	DE - 222	RND			
BF - 191	USING	CF - 207	*	DF - 223	LOG			
E0 - 224	EXP	F0 - 240	CSNG					
E1 - 225	COS	F1 - 241	CDBL					
E2 - 226	SIN	F2 - 242	FIX					
E3 - 227	TAN	F3 - 243	LEN					
E4 - 228	ATN	F4 - 244	STR$					
E5 - 229	PEEK	F5 - 245	VAL					
E6 - 230	CVI	F6 - 246	ASC					
E7 - 231	CVS	F7 - 247	CHR$					
E8 - 232	CVD	F8 - 248	LEFT$					
E9 - 233	EOF	F9 - 249	RIGHT$					
EA - 234	LOC	FA - 250	MID$					
EB - 235	LOF	FB - 251						
EC - 236	MKI$	FC - 252						
ED - 237	MKS$	FD - 253						
EE - 238	MKD$	FE - 254						
EF - 239	CINT	FF - 255						

Tabelle 4. BASIC-Tokens (TRS-80 M1)

6Ah. Aha, das könnte eine Adresse innerhalb des betrachteten Speicherbereiches sein — 6A64h. Auf 6A63h liegt das Trennbyte '00'. Die beiden nächsten Bytes sind wieder eine Adresse, die auf den Anfang der folgenden Programmzeile zeigt usw. ...

BASIC führt also in jedem Informationsblock (hier: Programmzeile) eine Adresse (Zeiger) mit, die auf den Anfang des nächsten Datenblockes verweist. Eine solche Datenstruktur nennt man übrigens eine '(einfach) verkettete Liste'. Jetzt wird auch die Ein-Byte-Differenz zwischen tatsächlichem Programmtext-Anfang und Inhalt des HP deutlich — der HP ist der Zeiger auf das erste Listen-Element:

Wenn Sie die Zeiger-Systematik einmal weiterverfolgen, finden Sie irgendwann als Zeiger auf die nächste Zeile die Bytefolge 00 00. Daran erkennt der Interpreter, daß hier der Programmtext endet — er hat also die letzte Anweisung abgearbeitet und führt jetzt automatisch END (impliziertes END) aus. GOTO oder RETURN ändern den Zeiger auf die nächste Zeile (Anweisung). (GOTO durch die angegebene Zeilennummer und RETURN durch die Adresse auf dem Stapel).

Bleiben nun noch drei Bytes übrig, die sondiert werden müssen. Sie werden vielleicht schon die Zeilennummern vermißt haben: Diese finden sich auf den Plätzen 4 und 5 der im Speicher niedergelegten Programmzeile. Für unser Beispiel lauten diese beiden Bytes 64h und 00h — also 0064h, was die Zeilennummer 100 repräsentiert.

Da sich mit zwei Bytes in vorzeichenloser Darstellung 65536 Zahlen darstellen lassen, ergibt sich als größte Zeilennummer (theoretisch) 65535. Ihrem Rechnerhandbuch werden Sie aber schließlich schon entnommen haben, daß diese größtmögliche Zahl tatsächlich aber gar nicht mehr als Zeilennummer akzeptiert wird, sondern daß der Zeilennummernbereich irgendwo vorher begrenzt ist, beispielsweise bei 65529. Wozu diese Begrenzung?

Nun, sie reserviert einige Zeilennummern für den Interpreter, so daß dieser intern bestimmte Sonderzustände kennzeichnen kann. So zum Beispiel, daß er sich im Direkt-Modus befindet, oder im Falle der ERL-Funktion, daß es keine Programmzeile gibt, in der ein Fehler aufgetreten ist.

Kommen wir zum letzten ungeklärten Byte des speicherinternen Formates unserer Programmzeile 100. Zu finden ist hier 93h — von der REM-Anweisung in Zeile 100 aber keine Spur ... Was sagt Tabelle 4? Aha, 93h ist das Token von 'REM', also beginnt mit dem sechsten Byte der Zeile unser Programmtext. — Hier in Form eines tokisierten Schlüsselwortes.

Bevor ich Ihnen noch einige Tips und Tricks zu den zum Programmtext-Segment gehörigen Pointern (HP und TP) gebe, eine kurze Zusammenfassung über die Struktur des hier besprochenen Speicherabschnittes:

Byte 1 : immer '00', markiert den Anfang jeder folgenden Programmzeile.

Bytes 2,3: Zeiger auf den Anfang der nächsten Programmzeile im Rechnerspeicher. Wenn '0000', liegt Programmtextende vor.

Bytes 4,5: binäres Äquivalent der Zeilennummer;

Bytes 6 — n: Programmtext in tokisierter Form.

Nun zu den Pointern: In Tabelle 2 sehen Sie einen Auszug des BASIC-Workspace (MICROSOFT-BASIC auf TRS-80/M1), in dem einige Adressen durch Kästchen markiert sind. Die im ersten Kästchen gekennzeichnete Adresse enthält in Übereinstimmung mit Tabelle 5 den HP, die zweite Markierung zeigt den TP. Der eine oder andere Leser (und TRS-80-Anwender) mag sich nun wundern, weil er ganz andere Adressen vorfindet (z. B. 42E9h im TRS-LEVEL2-BASIC), wenn er sich mit dem Monitor denselben Speicherbereich ansieht. Wer seinen TRS-80 mit DISK-BASIC unter NEWDOS/80 fährt, wird die gemachten Angaben dagegen bestätigen, andererseits hat der TRSDOS-Anwender schon wieder Einwände. Waren wir nicht von der Voraussetzung ausgegangen, daß sich der HP nicht ändert? Nun, prinzipiell hat das seine Richtigkeit, nur ist es — erstens — vom Betriebssystem abhängig (LEVEL-2, TRSDOS 1.1/1.3/2.3/2.3B, NEWDOS/40, NEWDOS/80, DOSPLUS, LDOS, CP/M und was es sonst noch alles für den TRS-80 gibt), auf welchen Wert der HP beim Einstieg ins BASIC gesetzt wird. Zweitens ist beim DISK-BASIC entscheidend, wie viele Puffer (0—15, je 290 Bytes) für das diskettenorientierte Input/Output beim Hochfahren des BASIC-Interpreters reserviert werden.

Es bedarf wohl keiner Diskussion, daß der HP nicht mehr geändert werden darf, wenn sich bereits ein Programm im Programmspeichersegment befindet. Gegen eine Verschiebung des HP spricht aber nichts, solange Sie noch kein Programm geladen haben. Durch 'Umpoken' des HP auf eine höhere Speicheradresse könnten Sie sich zum Beispiel einen bestimmten Speicherbereich reservieren, den Sie für Maschinenprogramme nutzen wollen. Welche Vorteile sich daraus allerdings im Vergleich zum standardmäßigen Reservieren von Speicherplatz am oberen Speicherende ergeben, mag dahingestellt bleiben.

```
# DUMP  6A45
6A40                  00 64 6A 64 00 93 20 2A 2A 2A 20  *     .djd.. *** *
6A50  50 72 6F 67 72 61 6D 6D 2D 41 6E 66 61 6E 67 20  *Programm-Anfang *
6A60  2A 2A 2A 00 93 6A 6E 00 93 20 2A 2A 2A 20 53 70  ****..jn.. *** Sp*
6A70  65 69 63 68 65 72 2D 53 65 74 75 70 20 66 75 65  *eicher-Setup fue*
6A80  72 3A 20 22 42 41 53 49 43 20 69 6E 74 65 72 6E  *r: "BASIC intern*
6A90  21 22 00 9E 6A 78 00 B8 20 31 30 30 30 00 A6 6A  *!"..jx.. 1000..j*
6AA0  B2 00 99 20 41 00 AE 6A BC 00 9A 20 43 00 B6 6A  *... A..j... C..j*
6AB0  96 00 9B 20 45 00 BE 6A A0 00 9B 20 47 00 D3 6A  *... E..j... G..j*
6AC0  AA 00 B0 20 BE 20 49 28 41 2C 42 29 20 D5 20 41  *... . I(A,B) . A*



6D80  9F 6D 90 01 41 33 28 58 2C 59 2C 5A 29 20 D5 20  *.m..A3(X,Y,Z) . *
6D90  32 D1 58 20 CD 20 32 D1 59 20 CD 32 D1 5A 00 AB  *2.X . 2.Y .2.Z..*
6DA0  6D 9A 01 B7 3A 20 B7 3A 20 B7 00 BA 6D A4 01 B2  *m...: .: ...m...*
6DB0  20 BE 20 49 28 35 2C 32 29 00 00 00              * . I(5,2)...    *
```

Tabelle 5. BASIC-Programm im Speicher

Ein Trick zum Schluß

Mit den vielen Giften ist es so: Richtig dosiert, sind sie ein gutes Heilmittel. Wenn also oben gesagt wurde, daß der HP nicht mehr geändert werden darf, wenn sich ein Programm im Rechner befindet, so ist aus diesem 'Gift' für BASIC-Interpreter, die keinen Befehl zum Nachladen (Anhängen) von Programmzeilen kennen, jedoch eine 'Theorie' entwickelbar. Wenn Sie bereits ein Programm geladen haben und dann den HP so ändern, daß er zwei Speicherplätze vor den TP zeigt, ist dieses Programm zunächst für den BASIC-Interpreter verdeckt. Zwischen HP und TP steht nur die Endmarkierung des Programmtextes — 'kein Programm vorhanden!' heißt das für den Interpreter. In dieser Situation laden wir für gewöhnlich ein Programm, und es spricht nichts dagegen, daß wir es nun, nach Verschiebung des HP, gleichfalls tun. Das neue Programm wird dann ab HP im Speicher abgelegt und der TP während des Ladens entsprechend nachgeführt. Dabei wird auch die alte Endmarkierung durch das nachgeladene Programm überschrieben, weil der 'verbogene' HP darauf zeigte. Ist das zweite Programm nun im Speicher, müssen wir den HP noch auf seinen ursprünglichen Wert zurücksetzen. Anzumerken ist allerdings, daß die erste Zeilennummer des zweiten Programmes größer sein muß als die letzte des ersten Programmes, das heißt, die Zeilennummern dürfen sich nicht überschneiden.

Dieser kleine Trick zeigt, daß Kenntnisse über die speicherinternen Strukturen zu Verfahren im Umgang mit BASIC führen können, über die Ihr BASIC-Handbuch keine Aufschlüsse (mehr) gibt. In der nächsten Ausgabe von c't wollen wir die anderen Speicher-Segmente eingehender durchleuchten. Zur Übung mag der geneigte Leser auf der BASIS der hier vermittelten Kenntnisse einmal versuchen, ein Programm zu schreiben, das die Programmzeilen eines bestehenden BASIC-Programmes umnumeriert. Eine gute Problemlösung sollte GOTO- und GOSUB-Anweisung berücksichtigen. Auch sollten Anfangszeile und Schrittweite festzulegen sein.

(Fortsetzung folgt)

Zahlen Sie zuviel Steuern?

Gesamtlohnsteuertabellen auf dem Spectrum 16K/48K

Walter Grotkasten

'Der Nettolohn ist nur die Provision für die Erarbeitung der Steuern' — so seufzen allerorts die Steuerzahler. Was Sie zu zahlen haben und ob die Berechnungen vom Finanzamt richtig durchgeführt wurden, das überprüft dieses Programm, das alle notwendigen Steuertabellen enthält. Es gibt Ihnen darüber Auskunft, wieviel Lohnsteuer der Arbeitgeber vom laufenden monatlichen Arbeitslohn einbehält, wieviel Kirchensteuer und Investitionshilfesonderabgabe Sie zu zahlen haben.

Je nach Auszahlungsart können Sie zwischen einer Monats-, Wochen- und Tageslohnsteuertabelle wählen; diese Tabellen werden aus den Einkommensteuertabellen (Grund- und Splittingtarif) entwickelt. Ihre individuellen Freibeträge, die in der Lohnsteuerkarte eingetragen sind, werden mit berücksichtigt.

Ferner haben Sie die Wahl zwischen einer 'Allgemeinen Lohnsteuertabelle' (z. B. Angestellte, Arbeiter) und einer 'Besonderen Lohnsteuertabelle' (z. B. Beamte, Richter, Berufssoldaten). Die Höchstwerte der drei Tabellen liegen bei:

— Monatslohnsteuertabelle 80 000 DM
— Wochenlohnsteuertabelle 19 000 DM
— Tageslohnsteuertabelle 2 700 DM.

Als Berechnungsgrundlage für die Tabellen sind folgende Freibeträge und Pauschbeträge vorausgesetzt:

— der Grundfreibetrag in den Steuerklassen 1, 2 und 4	DM	4 212
— der Grundfreibetrag in der Steuerklasse 3	DM	8 424
— der Arbeitnehmer-Freibetrag in den Steuerklassen 1 bis 5	DM	480
— der Werbungskosten-Pauschbetrag in den Steuerklassen 1 bis 5	DM	564
— der Sonderausgaben-Pauschbetrag in den Steuerklassen 1, 2 und 4	DM	270
in der Steuerklasse 3	DM	540
— Vorsorgespauschale bis zu den jeweiligen Höchstbeträgen abhängig von der Anzahl der Kinder, Steuerklasse und Beitragsbemessungsgrenze zur Rentenversicherung (1984	DM	62 400)
— der Haushaltsfreibetrag in der Steuerklasse 2 bei Arbeitnehmern mit mindestens einem Kind	DM	4 212
— der Kinderfreibetrag je Kind in den Steuerklassen 2 und 3	DM	432
in der Steuerklasse 4	DM	216
— ein Rundungsbetrag in der Steuerklasse 6	DM	18

Die Kirchensteuer wird in einem bestimmten, in den einzelnen Bundesländern verschiedenen Prozentsatz von der Lohnsteuer erhoben. Sind Kinder zu berücksichtigen, so kann man die Kirchensteuer nicht unmittelbar aus der Lohnsteuer berechnen. Es werden je nach Anzahl der Kinder von der Lohnsteuer noch bestimmte Beträge abgezogen.

Die Investitionshilfeabgabe wird ab einer bestimmten Höhe der im Jahr angefallenen zu zahlenden Einkommensteuer erhoben. Sie beträgt grundsätzlich 5%. Zur Vermeidung von Härten wurden jedoch bestimmte Übergangsbereiche gebildet, in denen der Prozentsatz geringer ist.

Menü und Microdrive

Nach Eingabe des Listings muß das Programm mit GOTO 2000 gestartet werden. Wird das Programm nun mit LOAD 'LOHN' von der Kassette in den Rechner geladen, so erscheint nach dem erfolgreichen Ladevorgang die Menü-Karte auf dem Bildschirm. Durch Aufruf des Menüpunktes 'Hinweise' erhält man eine Kurzbeschreibung des Programmes. Es werden nun im Dialogverfahren alle Daten abgefragt, die zur Berechnung nötig sind.

Soll das Programm auf ein Microdrive gespeichert werden, so muß nach Anschluß des Interface und des Microdrive die Zeile 2000 wie folgt geändert werden:

SAVE*"m";1;"LOHN" LINE 0

Mit dem Befehl

LOAD*"m";1;"LOHN" LINE 0

wird das Programm von dem Microdrive in den Rechner geladen.

Kurzbeschreibung

Zeile	
30:	Initialisierung der Variablen
35:	Aufbau der Menü-Karte
50:	Auswahl der Tabellenart (Monats-, Wochen- oder Tageslohnsteuertabelle)
70:	Allgemeine Hinweise für den Programmablauf (Auswahl durch die Menü-Karte)
93:	Eingabe, ob die 'Allgemeine Lohnsteuertabelle' oder die 'Besondere Lohnsteuertabelle' berechnet werden soll
100:	Eingabe der Steuerklasse 1 bis 6
130:	Ausgabe bei falscher Eingabe
140:	Eingabe der Anzahl der Kinder (max. 13 Kinder)
165:	Ausgabe bei falscher Eingabe
170:	Eingabe des Freibetrages (in der Steuerkarte eingetragen)
200:	Eingabe des Anfangswertes der Tabelle, die berechnet werden soll
265:	Eingabe des Endwertes der Tabelle, die berechnet werden soll
290:	Die letzten 3 Eingaben werden auf ihren Betrag überprüft
310:	Eingabe des Bundeslandes wo Sie wohnen
340:	Initialisierung für die Monatslohnsteuertabelle
360:	Initialisierung für die Wochenlohnsteuertabelle
380:	Initialisierung für die Tageslohnsteuertabelle
400:	Berechnung der Pauschbeträge (abhängig von der Wahl der Steuerklasse)
440:	Berechnung der Kirchensteuer
480:	Berechnung der Vorsorgepauschale
505:	Ausgabe für den Tabellenkopf
605:	Aufruf des Unterprogrammes: Berechnung der Einkommensteuertabelle
640:	Berechnung der Lohnsteuer
685:	Berechnung der Investitionshilfesonderabgabe
800:	Ausgabe der Daten für die Tabelle
810:	Überprüfung, ob Tabellenende erreicht wurde
825:	Überprüfung, ob Bildschirmseite gefüllt ist
1000:	Unterprogramm: Überprüfung der Eingaben
1500:	Unterprogramm: Berechnung der Einkommensteuertabellen

```
    1 REM
    2 REM       COPYRIGHT by
    3 REM Dipl.-Ing. W.Grotkasten
    4 REM Hoefinger Str.26
    5 REM 7250 Leonberg
    6 REM Tel. 07152/54878
    7 REM
   15 BRIGHT 1: CLS
   20 PRINT AT 7,7;"Berechnung  d
er",TAB 7;"■■■■■■■■■■■■■■■■■■■■■■": PRI
NT AT 14,6;"Lohnsteuertabelle","
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"
   25 PAUSE 200: CLS
   30 POKE 23658,8: LET EIN=1500:
 LET TEST=1000: LET NL=0: LET EI
=1: LET FU=5: LET COU=6: LET EB=
NL: LET DR=NL: LET AA=80000: LET
 AB=2700: LET AC=19000: LET S1=2
340: LET S2=1170: LET E$="A"
   35 PRINT AT 2,11;"M E N U E",T
AB 11;"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"
   40 PRINT AT 6,NL;"Waehlen Sie
aus:",,,,,,"- (M)onatslohnsteuert
abelle",,,,"- (W)ochenlohnsteuert
abelle",,,"- (T)ageslohnsteuerta
belle",,,"- (H)inweise"
   45 IF INKEY$="H" THEN GO TO 70
   50 IF INKEY$="M" THEN LET B$="
M": GO TO 93
   55 IF INKEY$="W" THEN LET B$="
W": GO TO 93
   60 IF INKEY$="T" THEN LET B$="
T": GO TO 93
   65 GO TO 45
   70 CLS : PRINT "Sie haben die
Wahl zwischen",,,"einer Monats,-
 Wochen und Tages-",,"lohnsteuer
tabelle. In diesen",,,"Tabellen
werden Ihre indivi-",,,"duellen
Freibetraege mit einge-",,,"arbe
itet. Nach den neuen Steuer-",,"
gesetzen gibt es eine ALLGEMEINE
",,"LOHNSTEUERTABELLE (Angestell
te)",,,"und eine BESONDERE LOHNS
TEUER-",,,"TABELLE (Beamte)."
   71 PRINT AT 21,NL;"(W)EITER"
   73 IF INKEY$<>"W" THEN GO TO 7
3
   75 CLS : PAUSE 50: PRINT ,,"Di
e Hoechstwerte der 3 Tabellen",,
,"liegen bei:",,,,,,"MONATSLOHNS
TEUERTABELLE 80000 DM",,"WOCHENL
OHNSTEUERTABELLE 19000 DM",,"TAG
ESLOHNSTEUERTABELLE   2700 DM":
PRINT AT 21,NL;"(W)EITER"
   76 IF INKEY$<>"W" THEN GO TO 7
5
   80 CLS : PRINT "In der Tabelle
 wird fuer die",,,"T a b e l l e
n a r t eine ",,,"Abkuerzung be
nutzt.": PRINT ,,TAB 10;"A/W/Z/3
": PRINT ,,"A ==> Allgemeine Loh
nsteuertab. oder",,"B ==> Besond
ere Lohnsteuertab.",,,"W ==> Woc
henlohnsteuertabelle","oder",,"M
 ==> Monatslohnsteuertabelle","o
der",,"T ==> Tageslohnsteuertabe
lle": PRINT ,,"Z ==> Steuerklass
e 2","3 ==> 3 Kinder."
   85 PRINT AT 21,NL;"X ==> Rueck
sprung ins Menue"
   90 IF INKEY$<>"X" THEN GO TO 9
0
   91 CLS : GO TO 30
   93 CLS : PRINT AT 3,NL;"Soll d
ie",,,,,,"- (A)LLGEMEINE LOHNSTE
UERTABELLE",,"    oder",,,,"- (B)
ESONDERE  LOHNSTEUERTABELLE",,,,
"berechnet werden?"
   94 IF INKEY$="A" THEN GO TO 10
0
   95 IF INKEY$="B" THEN LET E$="
B": LET S1=1000: LET S2=1000: GO
 TO 100
   96 GO TO 94
  100 CLS
  101 PRINT AT FU,NL;"Fuer welche
 Steuerklasse (1 - 6)",,"soll di
e Tabelle berechnet",,,"werden?"
  105 INPUT A$
  110 GO SUB TEST
  115 IF RR=EI THEN GO TO 105
  120 CLS : LET EA=INT VAL A$
  125 IF EA>=EI AND EA<=6 THEN GO
 TO 135
  130 PRINT AT 15,NL;"Falsche Ein
gabe. Bitte neu ein-",,,"geben.
Es gibt nur die Steuer-",,,"klas
se 1, 2, 3, 4, 5, 6.": GO TO 101
  135 CLS : IF EA=EI OR EA=FU OR
EA=6 THEN GO TO 170
  140 PRINT AT FU,NL;"Anzahl der
Kinder"
  145 INPUT A$
  150 GO SUB TEST
  155 IF RR=EI THEN GO TO 145
  160 LET EB=INT VAL A$: IF EB>13
 THEN GO TO 140
  165 CLS : IF EA=2 AND EB=NL THE
N PRINT AT 5,NL;"Diese Steuerkla
ssenkombination",,,"gibt es nach
dem neuen Steuer-",,,"gesetz ni
cht mehr.": BEEP 1,5: PAUSE 300:
 CLS : RUN 30
  170 PRINT AT FU,NL;"Sollten Sie
 einen Freibetrag",,,"haben, so
geben Sie ihn ein;",,,"andernfal
ls eine 0 eingeben."
  175 INPUT A$
  180 GO SUB TEST
  185 IF RR=EI THEN GO TO 175
  195 CLS : LET EC=INT VAL A$
  200 PRINT AT FU,NL;"Eingabe des
 Anfangwertes der",,,"Tabelle:"
  205 INPUT A$
  210 GO SUB TEST
  215 IF RR=EI THEN GO TO 205
  220 CLS : LET ED=INT VAL A$: LE
T MAX=AA
  225 IF B$<>"T" THEN GO TO 235
  230 LET MAX=AB
  235 IF B$<>"W" THEN GO TO 245
  240 LET MAX=AC
  245 IF MAX>=ED THEN GO TO 255
  250 PRINT AT 15,NL;"Anfangswert
 zu hoch!": BEEP 1,5: GO TO 200
  255 IF EC<=ED THEN GO TO 265
  260 PRINT AT FU,NL;"Freibetrag
groesser als Anfangs-",,"wert!":
 BEEP 1,5: PAUSE 200: CLS : GO T
O 200
  265 PRINT AT FU,NL;"Eingabe des
 Endwertes der",,,"Tabelle"
  270 INPUT A$
  271 CLS
  275 GO SUB TEST
  280 IF RR=EI THEN GO TO 265
  285 CLS : LET EE=INT VAL A$
  290 IF MAX>=EE THEN GO TO 300
  295 PRINT AT 15,NL;"Endwert ist
 zu gross!": BEEP 1,5: GO TO 265
  300 IF EE>=ED THEN GO TO 310
  305 PRINT AT 15,NL;"Endwert kle
iner als Anfangswert.Bitte neu e
ingeben!": BEEP 1,5: GO TO 265
  310 CLS : PRINT AT 2,NL;"Wohnen
 Sie in einer der folgen-",,,"de
n Bundeslaender:",,,"-Bayern",,,
,"-Baden-Wuerttemberg",,,"-Breme
n",,,,"-Hamburg",,,,,,"Eingabe J
oder N."
  315 IF INKEY$="" THEN GO TO 315
  320 CLS
  325 IF INKEY$="J" THEN LET EF=6
: GO TO 340
  330 IF INKEY$="N" THEN LET EF=9
: GO TO 340
  335 GO TO 310
  340 IF B$<>"M" THEN GO TO 360
  345 LET GI=4.5: LET GA=ED-GI
  350 IF GA<NL THEN LET GA=NL
  355 LET GA=(INT (GA/GI))*GI+5.9
9: LET GC=12
  360 IF B$<>"W" THEN GO TO 380
  365 LET GI=1.05: LET GA=ED-GI
  370 IF GA<NL THEN LET GA=NL
  375 LET GA=(INT (GA/GI))*GI+1.0
9: LET GC=360/7
  380 IF B$<>"T" THEN GO TO 400
  385 LET GI=0.15: LET GA=ED-GI
  390 IF GA<NL THEN LET GA=NL
  395 LET GA=(INT (GA/GI))*GI+0.1
9: LET GC=360
  400 LET GD=18
```

```
 405 IF EA=EI THEN LET GD=1314
 410 IF EA=2 THEN LET GD=5526+EB
*432
 415 IF EA=3 THEN LET GD=1584+EB
*432
 420 IF EA=4 THEN LET GD=1314+EB
*216
 425 IF EA=FU THEN LET GD=1044
 430 LET GG=EI
 435 IF EA=3 OR EA=FU OR EA=5 TH
EN LET GG=NL
 440 LET GH=NL
 445 IF EB=EI THEN LET GH=600
 450 IF EB=2 THEN LET GH=1560
 455 IF EB>2 THEN LET GH=1560+(E
B-2)*1800
 460 IF EA=4 THEN LET GH=GH/2
 465 IF EA<>EI THEN GO TO 475
 470 LET GE=51: LET GF=52
 475 IF EA<>2 THEN GO TO 485
 480 LET GE=51+EB*600: LET GF=52
+EB*300
 485 IF EA<>3 THEN GO TO 495
 490 LET GE=2*51+EB*600: LET GF=
2*52+EB*300
 495 IF EA<>4 THEN GO TO 505
 500 LET GE=51+EB*300: LET GF=52
+EB*150
 505 PRINT "ARBEITS- /LOHNSTEUER
/ SONDERAB- LOHN      /
/ GABE"
 510 PRINT
 515 PRINT "           /TABELLEN-
/ KIRCHEN-            /ART
/ STEUER"
 520 PRINT
 525 LET HA=(GA-EC)*GC
 530 IF EA=5 OR EA=6 THEN GO TO
520
 535 LET HB=HA
 540 IF HB>62400 THEN LET HB=624
00
 545 LET HB=HB*0.09
 550 LET HC=HB
 555 IF HB>GE THEN LET HC=GE
 560 IF HB>GF THEN GO TO 575
 565 LET HC=HC+HB
 570 GO TO 580
 575 LET HC=HC+GF
 580 LET HC=(INT (HC/54))*54
 585 IF GG=EI AND HC<300 THEN LE
T HC=300
 590 IF GG=NL AND HC<600 THEN LE
T HC=600
 595 LET HA=HA-HC-GD
 600 LET KA=HA
 605 GO SUB EIN
 610 LET HD=KE
 615 GO TO 660
 620 LET HA=HA-GD
 625 LET HA=(INT (HA/54))*54
 630 LET KA=HA*1.5
 635 GO SUB EIN
 640 LET HD=KE
 645 LET KA=HA*2.5
 650 GO SUB EIN
 655 LET HD=KE-HD
 656 IF HD<NL THEN LET HD=NL
 660 LET HE=(HD-GH)*EF/100
 665 IF HE<NL THEN LET HE=NL
 670 LET HE=(INT (HE/GC*100))/10
0
 675 LET HD=(INT (HD/GC*100))/10
0
 680 IF B$="M" THEN LET HD=(INT
(HD*10))/10
 685 LET HF=NL
 690 IF B$<>"M" THEN GO TO 725
 695 IF EA=3 THEN GO TO 715
 700 IF HD>1250 THEN LET HF=(INT
 (HD-1250))*0.35
 705 IF HD>1458.3 THEN LET HF=HD
*0.05
 710 GO TO 725
 715 IF HD>2500 THEN LET HF=(INT
 (HD-2500))*0.35
 720 IF HD>2916.6 THEN LET HF=HD
*0.05
 725 IF B$<>"W" THEN GO TO 760
 730 IF EA=3 THEN GO TO 750
 735 IF HD>291.66 THEN LET HF=(I
NT ((HD-291.66)/0.2))*0.07
 740 IF HD>340.2 THEN LET HF=HD*
0.05
 745 GO TO 760
 750 IF HD>583.33 THEN LET HF=(I
NT ((HD-583.33)/0.2))*0.07
 755 IF HD>680.5 THEN LET HF=HD*
0.05
 760 IF B$<>"T" THEN GO TO 795
 765 IF EA=3 THEN GO TO 785
 770 IF HD>44.53 THEN LET HF=(HD
-41.66)*0.35
 775 IF HD>48.6 THEN LET HF=HD*0
.05
 780 GO TO 795
 785 IF HD>83.33 THEN LET HF=(HD
-83.33)*0.35
 790 IF HD>97.21 THEN LET HF=HD*
0.05
 795 LET HF=(INT (HF*100))/100
 800 PRINT GA; TAB 10;HD; TAB 23
;HF
 802 PRINT TAB 10;E$;"/";B$;"/";
EA;"/";EB; TAB 23;HE
 805 PRINT
 810 IF GA>=EE THEN GO TO 855
 815 LET GA=GA+GI
 820 LET COU=COU+3
 825 IF COU<21 THEN GO TO 525
 830 PRINT AT 21,NL;"(D)RUCKEN
  (W)EITER     (M)ENUE"
 835 IF INKEY$="D" THEN GO SUB 1
700: COPY
 840 IF INKEY$="M" THEN CLS : PA
USE 50: RUN 30
 845 IF INKEY$="W" THEN LET COU=
NL: CLS : GO TO 525
 850 GO TO 830
 855 PRINT AT 21,NL;"(D)RUCKEN",
"(M)ENUE"
 860 IF INKEY$="D" THEN GO SUB 1
700: COPY
 865 IF INKEY$="M" THEN CLS : PA
USE 50: RUN 30
 870 GO TO 855
1000 LET RR=NL
1005 FOR X=EI TO LEN A$
1010 IF A$(X)>="0" AND A$(X)<="9
" THEN GO TO 1025
1015 PRINT AT 15,NL;"FALSCHE EIN
GABE.  BITTE NEU EIN-",,"GEBEN!"
: BEEP 1,5
1020 LET RR=EI
1025 NEXT X
1030 IF A$="" THEN LET RR=EI
1035 RETURN
1040 IF KA<=18000 THEN GO TO 159
5
1500 IF GG=EI THEN GO TO 1530
1505 IF EA=3 THEN GO TO 1520
1510 LET KA=(54*INT (KA/54))/2
1515 GO TO 1535
1520 LET KA=(108*INT (KA/108))/2
1525 GO TO 1535
1530 LET KA=54*INT (KA/54)
1535 IF KA<=4212 THEN GO TO 1605
1540 IF KA<=18000 THEN GO TO 159
5
1545 IF KA<=59999 THEN GO TO 158
0
1550 IF KA<=129999 THEN GO TO 15
55
1555 LET KE=INT (0.56*KA-14837)
1560 GO TO 1610
1565 LET KE=(KA-60000)/10000
1570 LET KE=INT ((((0.09*KE-5.45
)*KE+88.13)*KE+5040)*KE+20018)
1575 GO TO 1610
1580 LET KE=(KA-18000)/10000
1585 LET KE=((((3.05*KE-73.76)*K
E+695)*KE+2200)*KE+3034)
1590 GO TO 1610
1595 LET KE=INT (0.22*KA-926)
1600 GO TO 1610
1605 LET KE=NL
1606 IF EA=5 OR EA=6 THEN LET KE
=INT (0.22*KA)
1610 IF GG=NL THEN LET KE=KE+KE
1615 RETURN
1700 PRINT AT 21,NL;"
1705 RETURN
2000 SAVE "LOHN" LINE 0
```

Mischkulturen

Spectrum 48 K als Hobby-Gärtner

Udo Bartz

Daß Erdbeeren besonders gut munden, wenn zwischen ihnen Knoblauch oder Zwiebeln angepflanzt werden, ist versierten Hobby-Gärtnern schon länger bekannt. Diese sogenannten Mischkulturen bieten aber noch erheblich mehr Vorteile. Bei geschickter Kombination der Pflanzen halten diese sich sogar gegenseitig die Schädlinge vom Hals, man spart also Schädlingsbekämpfungsmittel. Trotz dieser verdächtig gesunden Anbaumethode braucht man aber keine Ertragseinbußen hinzunehmen. Der gezielte Einsatz von Mischkulturen erfordert allerdings einen hohen 'Verwaltungsaufwand', wenn man alle Vorteile optimal nutzen will. Hier ist der Computer in seinem Element.

Foto: Eduard Dietl, BAVARIA

Wenn Mischkulturen richtig geplant werden, kann man Ermüdungserscheinungen des Bodens, wie sie bei Monokulturen zwangsläufig auftreten, weitgehend vermeiden. Dadurch ist man nicht gezwungen, den Boden mit immer größeren Mengen Kunstdünger vollzupumpen, was ganz sicher allen Beteiligten, nämlich Mensch, Pflanze und Boden, zugute kommt.

Die richtige Mischung

Mischkulturen sind der Natur abgeguckt, denn dort gibt es kein Monopol einer einzigen Pflanzenart. Die Pflanzen, die sich in der Wildnis an ihrem Standort behaupten, müssen sich aber auch miteinander 'vertragen' können. Sie bilden eine Biozönose (griech.: Lebensgemeinschaft). Solche Harmonie wird auch in der Biozönose 'Garten' angestrebt. Sollen die oben genannten Vorteile genutzt werden, muß man auch im Garten auf gute Verträglichkeit der Pflanzen untereinander achten.

Die Pflanzen dürfen sich mit ihren Früchten und Blättern nicht gegenseitig behindern, sie müssen auch und gerade im

```
VORSCHLAG Nr. 1 (Beetbr.=1.20 m)
                        24/39 RAD./KR.
######################  21 KOHLRABI
++++++++++++++++++++++  23 KOPFSAL.
                        24/39 RAD./KR.
####################    20 KOHL
                        24/39 RAD./KR.
++++++++++++++++++++++  23 KOPFSAL.
######################  21 KOHLRABI
                        24/39 RAD./KR.
SPAETER KOENNEN NOCH WINTERSALA-
TE ODER SPINAT NACHGESAET WERDEN
```

Wurzelbereich miteinander harmonieren, und ihre Nährstoffentnahme muß berücksichtigt werden. Pflanzen geben auch eine Vielzahl von Substanzen an den Boden zurück, die von ihren Nachbarn nicht immer gut vertragen werden. So spielen die Düfte und Wurzelausscheidungen bei der Wahl ebenfalls eine wichtige Rolle.

Zu diesem Thema gibt es mittlerweile eine ganze Wissenschaft, die sich, gestützt auf jahrzehntelange Untersuchungen und Beobachtungen, mit diesen Wechselwirkungen beschäftigt. Die Ergebnisse sind verblüffend und einfach zugleich. Manche Pflanzen können sich buchstäblich nicht 'riechen' und gehen ein, wenn sie zu entsprechender Nachbarschaft gezwungen werden, andere Kombinationen blühen förmlich auf und wirken so günstig aufeinander ein, daß man nur staunen kann. 'Artenfreundliche' Pflanzen verflechten sich im Wurzelbereich sogar, während 'artenfeindliche' Nachbarn sich zumeist auf einen engen, isolierten Wurzelbereich zurückziehen und unter Umständen verkümmern.

Diese Beobachtungen sind gar nicht neu. Schon den alten Chinesen und anderen Kulturvölkern waren diese Zusammenhänge bekannt und wurden bei der Planung und Gestaltung der Gärten und Felder berücksichtigt. In das hier vorgestellte Programm sind diese Erkenntnisse eingebracht worden und können für 62 Pflanzenarten ausgewertet werden.

Mischen — Säen — Ernten

Die Struktur des Programms ist bewußt einfach und übersichtlich gestaltet, um individuelle Erweiterungen recht einfach vornehmen zu können. Bei der Planung Ihres Gartens gehen Sie am besten wie folgt vor:

Lassen Sie sich zunächst die Liste aller hier benutzten Pflanzen ausdrucken, und suchen Sie sich diejenigen heraus, die Sie anbauen wollen. (Falls Sie keinen Drucker haben, sehen Sie einfach im Listing — Zeile 8000 bis 8030 — nach.) Danach lassen Sie sich vom SPECTRUM zeigen, welche Pflanzen Sie auf keinen Fall auf ein gemeinsames Beet setzen dürfen. Unter Berücksichtigung der verschiedenen Pflanz- und Erntezeiten können Sie nun mit Hilfe des Programms Ihre Beete zusammenstellen. Sollten Sie noch ungeübt in diesem Metier sein, dann sehen Sie sich vielleicht zuerst die in drei ausgewählten Beispielen im Programm vorhandenen Computer-Vorschläge an.

Pflanz- und Erntezeiten sind im Programm selbst nicht berücksichtigt worden, da dies wegen der Vielfalt des erhältlichen Saatgutes kaum möglich ist.

Alle Jahre wieder

Denken Sie bitte daran, daß eine Mischkultur im Gegensatz zur herkömmlichen zeitlichen Fruchtfolge nicht ein Nacheinander, sondern ein Miteinander in einer Pflanzengemeinschaft, also eine räumliche Fruchtfolge, darstellt. Ein einmal festgelegtes Schema können und sollten Sie daher immer wieder benutzen, denn das bunte Gemisch ist nicht wahllos zusammengewürfelt, sondern folgt einem wohldurchdachten Plan, um die wechselseitig schützenden, vorbeugenden und fördernden Eigenschaften der Pflanzen bestmöglich auszunutzen.

Ein gutes Beispiel ist das Zusammenleben von Möhren und Zwiebeln: beide vertreiben sich gegenseitig mit ihren Ausscheidungen die Schädlinge (Möhren- und Zwiebelfliege). Aber auch auf den Geschmack der geernteten Früchte wirken sich Mischkulturen positiv aus.

Erweiterungen

Das Programm benötigt fast 21 KByte an Speicherplatz, und ist so aufgebaut, daß alle wichtigen Angaben in DATA-Zeilen vorhanden sind. Dies läßt den SPECTRUM sehr schnell reagieren und macht das Einfügen Ihrer eigenen Pflanzenkombinationen in das Programm sehr einfach. Dazu ein Beispiel:

Zwei neue Pflanzen sollen aufgenommen werden. Ihre Namen werden zuerst an das DATA-Feld ab Zeile 8000 angehängt, dann wird die Laufvariable der beiden Schleifen in den Zeilen 9 und 50 um zwei erhöht, da jetzt 64 Einträge vorhanden sind. Die Querverweisroutine in den Zeilen 6300 bis 6360 sollte komplett verschoben werden. Dazu eignet sich der Bereich 8300 bis 8360. Die Variable 'qv' in Zeile 11 wird entsprechend in 8300 geändert. So haben Sie Platz für sieben weitere Pflanzen. Da der Zugriff auf die einzelnen Pflanzen in Hunderterschritten erfolgt, (Zeile 52 und 67) verfahren Sie nach dem gleichen Muster und legen ähnliche Routinen in den Zeilen 6300 und 6400 für die beiden neu hinzugekommenen Pflanzenarten an, wie Sie sie in den anderen Hunderterblöcken erkennen können. Sie dürfen auch nicht vergessen, das Feld für A$ (Zeile 8) um die Anzahl der Neuzugänge zu erweitern, desgleichen die Variablen 'i' in der Zeile 35.

Grünes Menue

Über ein Menue haben Sie nach dem Programmstart die Möglichkeit, sich entweder die zueinander passenden Pflanzen herauszusuchen, oder die Vorschläge des SPECTRUM anzusehen, Ihre eigenen Kreationen zu entwickeln und ausdrucken zu lassen, oder sich eine komplette Liste der im Programm vorhandenen Pflanzen vom Drucker ausgeben zu lassen. Bei der Pflanzenauswahl können Sie kreuz und quer durch die Kombinationen springen, Sie brauchen nur die entsprechenden Codezahlen für die jeweilige Pflanze einzugeben.

Haben Sie den Menuevorschlag 'Eigene Beetgestaltung' gewählt, fragt der SPECTRUM nach Beetnummer oder Lagebezeichnung und Pflanzzeit. Dann ist die Anzahl der zu bepflanzenden Reihen einzugeben. Der Computer zeichnet daraufhin ein entsprechend großes Beet und links daneben einen Cursor, mit dem Sie herauf- und herunterfahren können, um die gewünschten Reihen mit den Pflanzennamen zu kennzeichnen oder um Korrekturen vorzunehmen. Mit den Tasten '6' und '7' ist der Cursor zu bewegen, und mit 'P' teilen Sie mit, daß Sie in die rechts neben dem Cursor stehende Reihe etwas eintragen wollen. Dieser Eintrag darf maximal 19 Zeichen lang sein, was ausreichen sollte, um auch zwei Namen gleichzeitig in eine Reihe zu bekommen.

Wenn Sie fertig sind, tippen Sie auf die Frage nach der Pflanzenart einfach ENTER. Nunmehr stellt der SPECTRUM Ihnen nacheinander die sieben ungünstigsten Kombinationen vor, wie sie in einer Mischkultur auf jeden Fall vermieden werden sollten, und fragt, ob diese in Ihrer Kreation vorhanden sind. Wird dies in irgendeinem Fall bejaht, erhalten Sie die Möglichkeit zur Korrektur. Ist alles okay, dann können Sie sich Ihr Beet ausdrucken lassen, um es in Ihren Gartenplan einzukleben. Vorsichtshalber sehen Sie aber noch einmal im Menuevorschlag '1' nach, ob alle von Ihnen gewählten Pflanzen sich miteinander vertragen.

Listing

Im Programmlisting bedürfen einige Passagen einer Erläuterung, da der verwendete Drucker nicht alle Zeichen des SPECTRUM kennt und es ihm daher beliebte, eigene Merkwürdigkeiten auszugeben:

Bei der gestrichelten Linie in der Zeile 20 handelt es sich nicht um das Minuszeichen, sondern um den roten Strich auf der Taste 0, der eine durchgehende Linie erzeugt. In der Variablen p$ in Zeile 7280 steht ein ¥, das durch das Copyright-Zeichen unter der Taste 'P' zu ersetzen ist. Gleiches gilt für die Zeile 7410. POKE 23658,8 sorgt dafür, daß der SPECTRUM ständig im Großbuchstabenmodus bleibt. Dies darf nicht geändert werden. In diesem Sinne also: 'Frohe Ernte'! □

Als weiterführende Literatur sei empfohlen:

DER BIO-GARTEN
von Marie-Luise Kreuter,
Büchergilde Gutenberg,
München 1981

GESUNDER GARTEN
DURCH MISCHKULTUR
von Gertrud Franck,
Südwest-Verlag 1980

```
  1 GO TO 8
  2 SAVE "FARMER" LINE 1
  8 DIM A$(62,15): POKE 23658,8
  9 RESTORE 8000: FOR n=1 TO 62
: READ a$(n): NEXT n
 11 LET 9v=6300: LET p=0: LET l
$="
  "
 15 CLS : PRINT "BITTE WAEHLEN:
"'''
 16 LET p=0
 20 PRINT ''"1 MISCHKULTURPFLAN
ZEN AUSSUCHEN ___________________
_____________2 COMPUTERVORSCHLA
EGE             ________________
_____________3 EIGENE BEETGESTA
LTUNG           ________________
_____________4 LISTE ALLER PFLA
NZEN(PRINTER) ___________________
_____________"
 25 LET e$=INKEY$: IF INKEY$="" 
THEN  GO TO 25
 30 IF e$<"1" OR e$>"4" THEN  G
O TO 25
 31 IF e$="1" THEN  GO TO 40
 32 IF e$="2" THEN  GO TO 7000
 34 IF e$="3" THEN  GO TO 9000
 35 IF e$="4" THEN  FOR i=1 TO
62
 36 LPRINT i;"->";a$(i): NEXT i
 38 GO TO 15
 40 CLS : INPUT "PFLANZE ? ";B$
: LET B$=(B$+"                 "
)( TO 15)
 50 RESTORE 8000: FOR N=1 TO 62
: READ a$(n)
 52 IF a$(n)=b$ THEN  PRINT AT
0,1;"Zu ";b$'"Passen in der Misc
hkultur:": LET p=100*n
 65 NEXT N
 66 IF p=0 THEN  PRINT "KEIN EI
NTRAG VORHANDEN.": PAUSE 100: GO
 TO 15
 67 GO SUB p
 99 GO TO 15
100 PRINT 20; PAPER 4; INK 9;a$
(20): GO SUB 9v: RETURN
200 RESTORE 290
205 READ bk: IF LEN STR$ bk=1 T
HEN  PRINT "0";
206 PRINT bk; PAPER 4; INK 9;a$
(bk): IF bk<>55 THEN  GO TO 205
208 PRINT "weitere Pflanzen:"
210 INPUT "weiter mit ENTER:";z
$
215 FOR i=2 TO 21: PRINT AT i,0
;l$: NEXT i
217 PRINT AT 1,0;l$;AT 1,0;"wei
ter Passen:"
219 READ bk: IF LEN STR$ bk=1 T
HEN  PRINT "0";
220 PRINT bk; PAPER 4; INK 9;a$
(bk): IF bk<>22 THEN  GO TO 219
230 PRINT "nicht dazuPassend:"
235 READ bk: PRINT bk; PAPER 2;
 INK 9;a$(bk): IF bk<>60 THEN  G
O TO 235
240 GO SUB 9v: RETURN
290 DATA 61,62,9,13,16,23,27,30
,37,46,39,40,43,48,52,55,15,8,36
,1,25,22
292 DATA 10,19,60
300 RESTORE 390
305 READ bo: IF LEN STR$ bo=1 T
HEN  PRINT "0";
310 PRINT bo; PAPER 4; INK 9;a$
(bo): IF bo<>29 THEN  GO TO 305
330 PRINT "Aber nicht:"
340 READ bo: IF LEN STR$ bo=1 T
HEN  PRINT "0";
350 PRINT bo; PAPER 2; INK 9;a$
(bo): IF bo<>60 THEN  GO TO 340
360 GO SUB 9v: RETURN
390 DATA 13,21,23,29,9,12,27,60
400 RESTORE 490
405 FOR i=1 TO 2: READ bk: PRIN
T bk; PAPER 4; INK 9;a$(bk): NEX
T i
420 GO SUB 9v: RETURN
490 DATA 60,61
500 PRINT 10; PAPER 4; INK 9;a$
(10)
510 GO SUB 9v: RETURN
600 PRINT 12; PAPER 4; INK 9;a$
(12)
610 GO SUB 9v: RETURN
700 PRINT 16; PAPER 4; INK 9;a$
(16)
710 GO SUB 9v: RETURN
800 RESTORE 890
820 READ di: IF LEN STR$ di=1 T
HEN  PRINT "0";
830 PRINT di; PAPER 4; INK 9;a$
(di): IF di<>60 THEN  GO TO 820
850 GO SUB 9v: RETURN
890 DATA 9,13,20,23,33,37,46,43
,60
900 RESTORE 990
910 READ er: IF LEN STR$ er=1 T
HEN  PRINT "0";
920 PRINT er; PAPER 4; INK 9;a$
(er): IF er<>8 THEN  GO TO 910
930 PRINT "Nicht aber:"
940 READ er: IF LEN STR$ er=1 T
HEN  PRINT "0";
950 PRINT er; PAPER 2; INK 9;a$
(er): IF er<>60 THEN  GO TO 940
960 GO SUB 9v: RETURN
990 DATA 12,13,20,23,33,39,40,5
8,29,8
995 DATA 61,19,27,55,60
1000 RESTORE 1090
1005 READ eb: IF LEN STR$ eb=1 T
HEN  PRINT "0";
1010 PRINT eb; PAPER 4; INK 9;a$
(eb): IF eb<>5 THEN  GO TO 1005
1015 PRINT "UnPassend sind:"
1020 READ eb: IF LEN STR$ eb=1 T
HEN  PRINT "0";
1030 PRINT eb; PAPER 2; INK 9;a$
(eb): GO SUB 9v: RETURN
1090 DATA 19,23,27,52,60,38,61,5
,20
1100 PRINT 12; PAPER 4; INK 9;a$
(12)'"--WINTERZWIEBELN "
1110 GO SUB 9v: RETURN
1200 RESTORE 1290
1210 READ fe: IF LEN STR$ fe=1 T
HEN  PRINT "0";
1220 PRINT fe; PAPER 4; INK 9;a$
(fe): IF fe<>46 THEN  GO TO 1210
1230 PRINT "NICHT KOMBINIEREN MI
T:"
1240 READ fe: PRINT fe; PAPER 2;
 INK 9;a$(fe): IF fe<>61 THEN  G
O TO 1240
1250 GO SUB 9v: RETURN
1290 DATA 13,11,57,59,6,38,37,23
,62,9,44,46,55,61
1300 RESTORE 1390
1310 READ 9u: IF LEN STR$ 9u=1 T
HEN  PRINT "0";
1320 PRINT 9u; PAPER 4; INK 9;a$
(9u): IF 9u<>51 THEN  GO TO 1310
1325 PRINT "ES PASSEN NICHT:"
1330 INPUT "weiter mit ENTER:";z
$: FOR i=1 TO 20: PRINT AT i,0;l
$: NEXT i
1340 PRINT AT 1,0;"PASSEN NICHT:
"
1350 READ 9u: PRINT 9u; PAPER 2;
 INK 9;a$(9u): IF 9u<>55 THEN  G
O TO 1350
1360 GO SUB 9v: RETURN
1390 DATA 29,3,9,8,48,60,27,43,2
3,20,12,25,22,61,19,50,51
1395 DATA 16,39,40,55
1400 RESTORE 1490
1410 READ hi: IF LEN STR$ hi=1 T
HEN  PRINT "0";
1420 PRINT hi; PAPER 4; INK 9;a$
(hi): IF hi<>9 THEN  GO TO 1410
1430 GO SUB 9v: RETURN
1490 DATA 19,61,9
1500 RESTORE 1590
1510 READ ka: PRINT ka; PAPER 4;
 INK 9;a$(ka): IF ka<>60 THEN  G
O TO 1510
1520 GO SUB 9v: RETURN
1590 DATA 20,27,48,60
1600 RESTORE 1690
1610 READ kt: IF LEN STR$ kt=1 T
HEN  PRINT "0";
1620 PRINT kt; PAPER 4; INK 9;a$
(kt): IF kt<>61 THEN  GO TO 1610
1630 PRINT "nicht zu KARTOFFELN
Passen:"
1640 READ kt: IF LEN STR$ kt=1 T
HEN  PRINT "0";
1650 PRINT kt; PAPER 2; INK 9;a$
(kt): IF kt<>55 THEN  GO TO 1640
1660 GO SUB 9v: RETURN
1690 DATA 20,21,52,29,7,31,54,17
,36,25,19,61
1695 DATA 43,50,48,13,9,55
1700 RESTORE 1790
1710 READ kk: PRINT kk; PAPER 4;
 INK 9;a$(kk): IF kk<>58 THEN  G
O TO 1710
1720 GO SUB 9v: RETURN
1790 DATA 16,39,40,55,58
1800 PRINT 23; PAPER 4; INK 9;a$
(23)
1810 GO SUB 9v: RETURN
1900 RESTORE 1990
1910 READ kn: PRINT kn; PAPER 4;
 INK 9;a$(kn): IF kn<>16 THEN  G
O TO 1910
1920 PRINT "keinesfalls aber:"
1930 READ kn: IF LEN STR$ kn=1 T
HEN  PRINT "0";
1940 PRINT kn; PAPER 2; INK 9;a$
(kn): IF kn<>51 THEN  GO TO 1930
1950 GO SUB 9v: RETURN
1990 DATA 41,28,56,10,34,14,55,4
3,33,13,16
1995 DATA 20,9,61,51
2000 RESTORE 2090
2010 READ ko: PRINT ko; PAPER 4;
 INK 9;a$(ko): IF ko<>30 THEN  G
O TO 2010
2020 INPUT "weiter mit ENTER:";z
$
2030 FOR i=1 TO 21: PRINT AT i,0
;l$: NEXT i
2040 PRINT AT 1,0;"weitere Pflan
zen:"
2050 READ ko: IF LEN STR$ ko=1 T
HEN  PRINT "0";
2055 PRINT ko; PAPER 4; INK 9;a$
(ko): IF ko<>61 THEN  GO TO 2050
2060 PRINT "jedoch nicht:"
2065 READ ko: PRINT ko; PAPER 2;
 INK 9;a$(ko): IF ko<>19 THEN  G
O TO 2065
2070 GO SUB 9v: RETURN
2090 DATA 16,48,43,55,52,37,23,6
2,27,15,36,25,22,39,40,46,30,9,8
,1,13,61
2095 DATA 60,10,19
2100 RESTORE 2190
2105 READ kr: IF LEN STR$ kr=1 T
HEN  PRINT "0";
2110 PRINT kr; PAPER 4; INK 9;a$
(kr): IF kr<>62 THEN  GO TO 2105
2120 GO SUB 9v: RETURN
2190 DATA 3,52,51,48,47,43,39,40
,27,23,16,9,21,62
2200 RESTORE 2290
2210 READ ki: PRINT ki; PAPER 4;
 INK 9;a$(ki): IF ki<>43 THEN  G
O TO 2210
2220 GO SUB 9v: RETURN
2290 DATA 13,20,43
2300 RESTORE 2390
2310 READ ks: PRINT ks; PAPER 4;
 INK 9;a$(ks): IF ks<>27 THEN  G
O TO 2310
2315 INPUT "weiter mit ENTER:";z
$
2320 FOR i=1 TO 21: PRINT AT i,0
```

```
;l$: NEXT i
2325 PRINT AT 1,0;"ausserdem Pas
st er zu:"
2330 READ ks: IF LEN STR$ ks=1 T
HEN  PRINT "0";
2335 PRINT ks; PAPER 4; INK 9;a$
(ks): IF ks<>52 THEN  GO TO 2330
2340 PRINT "nicht aber zu:"
2345 READ ks: PRINT ks; PAPER 2;
 INK 9;a$(ks): IF ks<>48 THEN  G
O TO 2345
2350 GO SUB 9v: RETURN
2390 DATA 39,40,24,18,33,13,10,6
0,57,55,51,47,43,29,27,3,21,20,1
2,61,36,8,9,52
2395 DATA 35,48
2400 RESTORE 2490
2405 READ ke: PRINT ke; PAPER 4;
 INK 9;a$(ke): IF ke<>40 THEN  G
O TO 2405
2410 GO SUB 9v: RETURN
2490 DATA 23,39,40
2500 RESTORE 2590
2505 READ ku: PRINT ku; PAPER 4;
 INK 9;a$(ku): IF ku<>43 THEN  G
O TO 2505
2510 GO SUB 9v: RETURN
2590 DATA 13,16,20,43
2600 RESTORE 2690
2605 READ kb: PRINT kb; PAPER 4;
 INK 9;a$(kb)
2610 PRINT "nicht aber:"
2615 READ kb: PRINT kb; PAPER 2;
 INK 9;a$(kb)
2620 GO SUB 9v: RETURN
2690 DATA 29,16
2700 RESTORE 2790
2705 READ la: PRINT la; PAPER 4;
 INK 9;a$(la): IF la<>13 THEN  G
O TO 2705
2710 PRINT "ER VERTRAEGT SICH NI
CHT MIT:"
2715 READ la: IF LEN STR$ la=1 T
HEN  PRINT "0";
2720 PRINT la; PAPER 2; INK 9;a$
(la): IF la<>61 THEN  GO TO 2715
2730 GO SUB 9v: RETURN
2790 DATA 60,48,33,55,47,23,21,2
0,10,62,15,13
2795 DATA 43,9,61
2800 PRINT 19; PAPER 4; INK 9;a$
(19)
2810 GO SUB 9v: RETURN
2900 RESTORE 2990
2905 READ ma: IF LEN STR$ ma=1 T
HEN  PRINT "0";
2910 PRINT ma; PAPER 4; INK 9;a$
(ma): IF ma<>9 THEN  GO TO 2905
2915 PRINT "Nicht jedoch:"
2920 READ ma: PRINT ma; PAPER 2;
 INK 9;a$(ma): IF ma<>43 THEN  G
O TO 2920
2930 GO SUB 9v: RETURN
2990 DATA 55,23,3,13,58,26,32,16
,9
2995 DATA 48,43
3000 RESTORE 3090
3005 READ mn: PRINT mn; PAPER 4;
 INK 9;a$(mn): IF mn<>33 THEN  G
O TO 3005
3010 GO SUB 9v: RETURN
3090 DATA 39,40,20,33
3100 RESTORE 3190
3105 READ me: PRINT me; PAPER 4;
 INK 9;a$(me): IF me<>34 THEN  G
O TO 3105
3110 GO SUB 9v: RETURN
3190 DATA 16,34
3200 PRINT 29; PAPER 4; INK 9;a$
(29)
3210 GO SUB 9v: RETURN
3300 RESTORE 3390
3310 READ mo: IF LEN STR$ mo=1 T
HEN  PRINT "0";
3315 PRINT mo; PAPER 4; INK 9;a$
(mo): IF mo<>23 THEN  GO TO 3310
3320 GO SUB 9v: RETURN
3390 DATA 60,55,57,39,40,30,27,1
9,8,46,45,47,42,44,23
3400 RESTORE 3490
3405 READ ob: PRINT ob; PAPER 4;
 INK 9;a$(ob): IF ob<>52 THEN  G
O TO 3405
3420 GO SUB 9v: RETURN
3490 DATA 31,19,52
3500 RESTORE 3590
3505 READ Pe: PRINT Pe; PAPER 4;
 INK 9;a$(Pe): IF Pe<>55 THEN  G
O TO 3505
3510 PRINT "nicht aber zu:"
3515 READ Pe: PRINT Pe; PAPER 2;
 INK 9;a$(Pe)
3520 GO SUB 9v: RETURN
3590 DATA 51,55,23
3600 RESTORE 3690
3605 READ Pf: PRINT Pf; PAPER 4;
 INK 9;a$(Pf): IF Pf<>23 THEN  G
O TO 3605
3610 GO SUB 9v: RETURN
3690 DATA 16,20,23
3700 RESTORE 3790
3705 READ Ps: IF LEN STR$ Ps=1 T
HEN  PRINT "0";
3710 PRINT Ps; PAPER 4; INK 9;a$
(Ps): IF Ps<>61 THEN  GO TO 3705
3720 GO SUB 9v: RETURN
3790 DATA 33,55,51,47,43,39,40,2
0,12,8,61
3800 PRINT 12; PAPER 4; INK 9;a$
(12)
3810 GO SUB 9v: RETURN
3900 RESTORE 3990
3905 READ ra: IF LEN STR$ ra=1 T
HEN  PRINT "0";
3910 PRINT ra; PAPER 4; INK 9;a$
(ra): IF ra<>55 THEN  GO TO 3905
3915 PRINT "sie 9ehoeren nicht z
wischen:"
3920 READ ra: PRINT ra; PAPER 4;
 INK 9;a$(ra)
3925 GO SUB 9v: RETURN
3990 DATA 24,3,20,9,10,21,23,30,
33,37,46,52,55,13
4000 PRINT "siehe unter RADIESCH
EN (Nr.39)"
4020 GO SUB 9v: RETURN
4100 PRINT 19; PAPER 4; INK 9;a$
(19);AT 2,13;"und"
4102 PRINT 54; PAPER 4; INK 9;a$
(54)
4105 GO SUB 9v: RETURN
4200 RESTORE 4290
4205 FOR i=1 TO 2: READ rm: PRIN
T rm; PAPER 4; INK 9;a$(rm): NEX
T i
4220 GO SUB 9v: RETURN
4290 DATA 33,44
4300 RESTORE 4390
4305 READ rb: IF LEN STR$ rb=1 T
HEN  PRINT "0";
4310 PRINT rb; PAPER 4; INK 9;a$
(rb): IF rb<>55 THEN  GO TO 4305
4320 PRINT "Nicht Jedoch zu:"
4325 READ rb: PRINT rb; PAPER 2;
 INK 9;a$(rb): IF rb<>29 THEN  G
O TO 4325
4330 GO SUB 9v: RETURN
4390 DATA 60,20,21,37,19,13,8,25
,22,61,23,52,55
4395 DATA 27,29
4400 RESTORE 4490
4410 FOR i=1 TO 3: READ sa: PRIN
T sa; PAPER 4; INK 9;a$(sa): NEX
T i
4420 GO SUB 9v: RETURN
4490 DATA 12,33,42
4500 PRINT "aehnlich wie LAUCH,
siehe Nr.27:"
4505 GO SUB 9v: RETURN
4600 PRINT "siehe bei PFLUECKSAL
AT: Nr.37   nach !!!"
4605 GO SUB 9v: RETURN
4700 RESTORE 4790
4705 READ sw: PRINT sw; PAPER 4;
 INK 9;a$(sw): IF sw<>33 THEN  G
O TO 4705
4720 GO SUB 9v: RETURN
4790 DATA 27,23,37,21,33
4800 RESTORE 4890
4805 READ se: IF LEN STR$ se=1 T
HEN  PRINT "0";
4810 PRINT se; PAPER 4; INK 9;a$
(se): IF se<>15 THEN  GO TO 4805
4820 PRINT "sie soll nicht zu:"
4825 READ se: PRINT se; PAPER 2;
 INK 9;a$(se): IF se<>29 THEN  G
O TO 4825
4830 GO SUB 9v: RETURN
4890 DATA 55,61,27,21,20,2,13,15
,16,23,29
4900 PRINT 20; PAPER 4; INK 9;a$
(20)
4905 GO SUB 9v: RETURN
5000 RESTORE 5090
5010 READ so: PRINT so; PAPER 4;
 INK 9;a$(so)
5020 PRINT "ABER NICHT ZU:"
5025 READ so: PRINT so; PAPER 2;
 INK 9;a$(so)
5030 GO SUB 9v: RETURN
5090 DATA 13,16
5100 RESTORE 5190
5105 READ sP: PRINT sP; PAPER 4;
 INK 9;a$(sP): IF sP<>23 THEN  G
O TO 5105
5110 PRINT "Passt nicht zu:"
5120 FOR i=1 TO 2: READ sP: PRIN
T sP; PAPER 2; INK 9;a$(sP): NEX
T i
5130 GO SUB 9v: RETURN
5190 DATA 55,35,13,37,23,19,60
5200 RESTORE 5290
5205 READ si: PRINT si; PAPER 4;
 INK 9;a$(si): IF si<>34 THEN  G
O TO 5205
5220 GO SUB 9v: RETURN
5290 DATA 10,43,55,53,39,40,21,2
0,16,34
5300 PRINT "siehe (mit Einschrae
nkun9en bez.Hoehe und Platz) unt
er BOHNEN   (Nr.03) nach"
5310 GO SUB 9v: RETURN
5400 RESTORE 5490
5405 PRINT AT 0,0;"  ";AT 2,0;"a
ls Schaedlin9sbekaempfer 9ut zu"
5410 FOR i=1 TO 3: READ ta: PRIN
T ta; PAPER 4; INK 9;a$(ta): NEX
T i
5420 GO SUB 9v: RETURN
5490 DATA 41,16,55
5500 RESTORE 5590
5505 READ to: PRINT to; PAPER 4;
 INK 9;a$(to): IF to<>17 THEN  G
O TO 5505
5510 PRINT "nicht Passend sind:"
5515 INPUT "weiter mit ENTER:";z
$
5520 FOR i=1 TO 21: PRINT AT i,0
;l$: NEXT i
5525 PRINT AT 1,0;"Passen nicht:
"
5530 READ to: IF LEN STR$ to=1 T
HEN  PRINT "0";
5535 PRINT to; PAPER 2; INK 9;a$
(to): IF to<>9 THEN  GO TO 5530
5540 GO SUB 9v: RETURN
5590 DATA 33,52,48,43,39,40,57,3
7,35,23,29,27,21,20,19,17
5595 DATA 16,13,12,9
5600 PRINT 19; PAPER 4; INK 9;a$
(19): PRINT "zur Schaedlin9sabwe
hr"
5605 GO SUB 9v: RETURN
5700 RESTORE 5790
5705 READ zi: PRINT zi; PAPER 4;
 INK 9;a$(zi): IF zi<>17 THEN  G
O TO 5705
5710 GO SUB 9v: RETURN
```

```
5790 DATA 55,53,23,33,12
5800 RESTORE 5890
5805 FOR i=1 TO 4: READ zu: PRIN
T zu; PAPER 4; INK 9;a$(zu): NEX
T i
5820 GO SUB 9v: RETURN
5890 DATA 60,53,29,17
5900 PRINT ''"9ehoert zu den ZIC
HORIE-Salaten siehe unter Nr.57:
"
5910 GO SUB 9v: RETURN
6000 RESTORE 6090
6005 READ zw: IF LEN STR$ zw=1 T
HEN  PRINT "0";
6010 PRINT zw; PAPER 4; INK 9;a$
(zw): IF zw<>4 THEN  GO TO 6005
6015 PRINT "nicht zusammenPflanz
en mit:"
6020 READ zw: IF LEN STR$ zw=1 T
HEN  PRINT "0";
6025 PRINT zw; PAPER 2; INK 9;a$
(zw): IF zw<>9 THEN  GO TO 6020
6040 GO SUB 9v: RETURN
6090 DATA 33,43,57,23,13,10,8,15
,4
6095 DATA 20,3,9
6100 PRINT "GRUNDSAETZLICH GILT
GLEICHES WIEZU BOHNEN (Nr.03) U.
 STANGEN-   BOHNEN (Nr.53) GESAG
T WURDE !"
6110 GO SUB 9v: RETURN
6200 RESTORE 6290
6210 FOR i=1 TO 3: READ en: PRIN
T en; PAPER 4; INK 9;a$(en): NEX
T i
6220 GO SUB 9v: RETURN
6290 DATA 12,27,20
6299 STOP
6300 INPUT "SOLL ICH UNTER DEN G
ENANNTEN AR-TEN WEITER NACHSEHEN
 (J/N)?";z$: IF z$="N" THEN  RET
URN
6310 INPUT "unter welcher Nummer
(2-stellig) "; LINE f$: IF LEN f
$<>2 THEN  GO TO 6310
6320 IF f$(1)<"0" OR f$(1)>"9" T
HEN  GO TO 6310
6322 IF f$(2)<"0" OR f$(2)>"9" T
HEN  GO TO 6310
6325 IF VAL f$>62 THEN  GO TO 63
10
6330 CLS : PRINT "Zu ";a$(VAL f$
);"Passen:"
6350 GO SUB VAL f$*100
6360 RETURN
7000 CLS : PRINT "ES FOLGEN 3 BE
ISPIELE FUER GUTE MISCHKULTUREN.
                    DIE NUMMERN EN
TSPRECHEN DEN CODEZAHLEN IM PROG
RAMM."
7010 INPUT "weiter mit ENTER:";z
$
7020 CLS : PRINT "VORSCHLAG Nr.
1 (Beetbr.=1.20 m)"
7030 PLOT 6,160: DRAW 180,0: DRA
W 0,-140: DRAW -180,0: DRAW 0,14
0
7040 PRINT AT 20,0;"PFLANZEN SIE
 IM FRUEHJAHR IN DERMITTE DES BE
ETES EINE REIHE KOHL": PAUSE 100
7042 LET end=20: LET r1=10: LET
r2=r1: LET p$="$20": GO SUB 7900
7055 PRINT AT 20,0;"RECHTS U. LI
NKS DAVON JE EINE GEMISCHTE REIH
E RADIESCHEN/KRESSE ": PAUSE 200
7057 LET end=17: LET r1=8: LET r
2=12: LET p$="~24/39": GO SUB 79
00
7065 PRINT AT 20,0;"DANEBEN JE E
INE REIHE KOPFSALAT ";l$: PAUSE
200
7070 LET end=20: LET r1=6: LET r
2=14: LET p$="+23": GO SUB 7900
7080 PRINT AT 20,0;"ES FOLGEN JE
 EINE REIHE KOHLRABIU. GEMISCHT:
KRESSE/RADIESCHEN    "
7085 LET end=20: LET r1=4: LET r
2=16: LET p$="#21": GO SUB 7900
7090 LET end=17: LET r1=2: LET r
2=18: LET p$="~24/39": GO SUB 79
00
7100 LET d$="RAD./KR.": PRINT AT
 2,24;d$;AT 8,24;d$;AT 12,24;d$;
AT 18,24;d$
7110 PRINT AT 4,24;"KOHLRABI";AT
 16,24;"KOHLRABI"
7115 PRINT AT 6,24;"KOPFSAL.";AT
 14,24;"KOPFSAL."
7120 PRINT AT 10,24;"KOHL"
7130 PRINT AT 20,0;"SPAETER KOEN
NEN NOCH WINTERSALA-TE ODER SPIN
AT NACHGESAET WERDEN"
7140 INPUT "VORSCHL. DRUCKEN ?(J
/N)";z$: IF z$<>"J" AND z$<>"N"
THEN  GO TO 7140
7150 IF z$="J" THEN  COPY : FOR
i=1 TO 5: LPRINT : NEXT i
7200 CLS : PRINT "VORSCHLAG Nr.2
(Beetbr. 1.20 m)"
7210 PLOT 6,160: DRAW 160,0: DRA
W 0,-64: DRAW -160,0: DRAW 0,64
7220 LET end=19: LET r1=4: LET r
2=8: LET p$="%": GO SUB 7900
7230 LET r1=2: LET r2=6: LET p$=
"↑": GO SUB 7900
7240 PRINT AT 2,21;"MOEHREN(33)"
;AT 6,21;"MOEHREN(33)"
7245 PRINT AT 4,21;"ZWIEBEL(60)"
;AT 8,21;"ZWIEBEL(60)"
7250 PRINT AT 10,0;"DIESE KOMBIN
ATION SCHUETZT SICH GEGENSEITIG
VOR SCHAEDLINGEN.    IM SPAETSOMM
ER WERDEN DANN AUF  DEM GLEICHEN
 BEET AUSGESAET:"
7260 PAUSE 200: LET p$="@": GO S
UB 7900
7270 PRINT AT 2,21;"FELDSAL(11)"
;AT 6,21;"FELDSAL(11)"
7280 LET r1=4: LET r2=8: LET p$=
"¥": GO SUB 7900
7290 PRINT AT 4,21;"SPINAT (52)"
;AT 8,21;"SPINAT (52)"
7300 INPUT "VORSCHLAG DRUCKEN (J
/N)?";z$
7305 IF z$="J" THEN  PRINT AT 2,
1;"FRUEH:MOEHR;SPAET->";AT 4,1;"
FRUEH:ZWIEB;SPAET->": COPY : FOR
 i=1 TO 5: LPRINT : NEXT i
7310 IF z$<>"J" AND z$<>"N" THEN
  GO TO 7300
7350 CLS : PRINT "AUCH MIT KARTO
FFELN LASSEN SICH GUTE MISCHKULT
UREN BILDEN:"''"VORSCHLAG Nr.3 (
EBENFALLS FUER  EINE BEETBREITE
VON 1.20 MTR.):"''"FRUEHKARTOFFE
LN, SPINAT, RADIES-CHEN, KAPUZIN
ERKRESSE, GRUENKOHLUND LAUCH:"
7360 INPUT "ENTER TIPPEN:";z$
7400 CLS : PRINT "VORSSCHLAG Nr.
3:"
7410 PRINT AT 2,0; PAPER 4; INK
0;"KAPUZINERKRESSE(17)~~~~~~~~~~~~
~~~";l$;"RADIESCHEN(39)..........
..........";l$;"KARTOFFELN(16)ooo
ooooooooooooooo";l$;"SPINAT(52)¥
¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥¥";l$;"KARTO
FFELN(16)oooooooooooooooooooo";l$;
"RADIESCHEN(39)...................
.";l$;"KAPUZINERKRESSE(17)~~~~~~~
~~~~~~~~"
7450 PRINT "->KARTOFFELN BRAUCHE
N NICHT AN-    GEHAEUFELT WERDEN.
<-  Nr. 17  IST SALAT UND KARTOF
FELSCHUTZ   GLEICHZEITIG.NACH ER
NTE VON 16  KANN GRUENKOHL UND L
AUCH GE-    PFLANZT WERDEN."
7455 INPUT "VORSCHLAG DRUCKEN (J
/N)?";z$: IF z$="J" THEN  COPY :
 FOR i=1 TO 5: LPRINT : NEXT i
7899 GO TO 15
7900 FOR i=1 TO end: PRINT AT r1
,i; PAPER 4; INK 0;p$;AT r2,i;p$
: PAUSE 20: NEXT i
7910 INPUT "ENTER TIPPEN";z$: RE
TURN
8000 DATA "BEIFUSS","BLUMENKOHL"
,"BOHNEN","BOHNENKRAUT","BORETSC
H","CHICOREE","DICKE BOHNEN","DI
LL","ERBSEN","ERDBEEREN","FELDSA
LAT","FENCHEL","GURKEN","HIMBEER
EN","KAMILLE","KARTOFFELN","KAPU
ZINERKRESSE","KERBEL","KNOBLAUCH
","KOHLARTEN","KOHLRABI","KORIAN
DER"
8010 DATA "KOPFSALAT","KRESSE","
KUEMMEL","KUERBIS","LAUCH","LILI
EN","MAIS","MANGOLD","MEERRETTIC
H","MELONEN","MOEHREN","OBSTBAEU
ME","PETERSILIE","PFEFFERMINZE",
"PFLUECKSALAT","RADICCHIO","RADI
ESCHEN","RETTICH","ROSEN"
8020 DATA "ROSMARIN","ROTE BETE"
,"SALBEI","SCHNITTLAUCH","SCHNIT
TSALAT","SCHWARZWURZELN","SELLER
IE","SENF","SONNENBLUMEN","SPARG
EL","SPINAT","STANGENBOHNEN","TA
GETES","TOMATEN","TULPEN","ZICHO
RIE","ZUCCHINI","ZUCKERHUT","ZWI
EBELN","BUSCHBOHNEN"
8030 DATA "ENDIVIEN"
9000 CLS : PRINT "VERMEIDEN SIE
AUF JEDEN FALL    FOLGENDE KOMBI
NATIONEN:"
9010 PRINT 'a$(12);"<->";a$(55)'
a$(61);"<->";a$(60)'a$(20);"<->"
;a$(60)'a$(55);"<->";a$(9)'a$(9)
;"<->";a$(3)'a$(16);"<->";a$(50)
'a$(20);"<->";a$(49)
9020 INPUT "ENTER TIPPEN";z$
9030 CLS : INPUT "BEET Nr. ODER
LAGE?";g$: PRINT "BEET: ";g$;
9040 INPUT "PFLANZZEIT ?";h$: PR
INT "(";h$;")"
9050 INPUT "WIEVIELE REIHEN (max
.:15)?";y$
9052 FOR i=1 TO LEN y$: IF y$(i)
<"0" OR y$(i)>"9" THEN  GO TO 90
50
9053 NEXT i
9054 LET y=VAL y$
9055 IF y<1 OR y>15 THEN  GO TO
9050
9060 PLOT 20,175-((y*8)+19): DRA
W 160,0: DRAW 0,(8*y+4): DRAW -1
60,0: DRAW 0,-(8*y+4)
9070 LET u$="->": LET x=2
9080 PRINT #0;AT 0,0;"REIHENWAHL
 mit:6/7 (P=FERTIG)"
9090 LET x$=INKEY$
9095 LET z=x
9100 LET x=x+(INKEY$="6" AND x<=
y)-(INKEY$="7" AND x>=3)
9105 PRINT AT z,0;"  ";AT x,0;u$
9110 IF x$="P" THEN  GO TO 9150
9115 PAUSE 5: GO TO 9090
9150 INPUT "WELCHE PFLANZE(N) ?"
;s$: IF s$="" THEN  GO TO 9200
9160 LET s$=(s$+"
     "X TO 19): PRINT AT x,3;s$
9170 GO TO 9080
9200 PRINT AT 18,0;"BITTE PRUEFE
N,OB FOLGENDE FEHL- KOMBINATIONE
N VORHANDEN SIND:"
9250 RESTORE 9500
9260 READ fa,fb: PRINT AT 21,0;
PAPER 2; INK 9;a$(fa);a$(fb)
9270 INPUT "VORHANDEN ?(J/N)";z$
: IF z$="J" THEN  GO TO 9070
9280 IF fb<>49 THEN  GO TO 9260
9290 INPUT "BEET DRUCKEN ?(J/N)"
;z$
9300 IF z$="J" THEN  FOR i=18 TO
 21: PRINT AT i,0;l$: NEXT i: CO
PY : FOR i=1 TO 5: LPRINT : NEXT
 i
9350 PRINT AT 10,8;"PROGRAMMENDE
";AT 11,8;"(RUN=RESTART)"
9500 DATA 12,55,61,60,20,60,55,9
,9,3,16,50,20,49
9510 STOP
```

Rolf Keller

I/O-Techniken:

Drucken per Interrupt

Die Verwendung von Interrupts ist eine äußerst elegante Methode, Peripheriegeräte schnell und effizient zu bedienen. Die CPU vertrödelt keine Zeit in Warteschleifen oder durch Statusabfragen, wenn sie aber gebraucht wird, dann ist sie da. Da fragt man sich doch, warum diese wundervolle Methode nicht ausschließlich angewendet wird. Nun, sie stellt bestimmte Anforderungen an die verwendete Hardware und Software. Auch mit Interrupts bestens vertraute Programmierer gewinnen dabei des öfteren noch 'völlig neue' Erkenntnisse.

Ein Interrupt (Unterbrechung) ist eine Programmverzweigung, üblicherweise in Form eines Unterprogrammaufrufes. Ähnlich wie bei einem GOSUB in BASIC-Programmen oder einem CALL in Assembler, wird ein Unterprogramm ausgeführt und in das übergeordnete, normalerweise aufrufende Programm, mit einem RETURN genau dorthin zurückgesprungen, wo es verlassen wurde. Eine derartige Verzweigung erfolgt per Interrupt aber, ohne daß im laufenden Programm ein diesbezüglicher Aufruf programmiert worden ist. Wie das?

Die CPU verfügt hardwaremäßig über (mindestens) eine Interrupt-Request-Leitung (Interrupt-Anforderung). Wird diese Leitung auf das (je nach Prozessor) für eine Interruptanforderung vorgeschriebene Potential gesteuert, führt die CPU eine Programmverzweigung auf eine ganz bestimmte Adresse durch. Diese Adresse ist prozessorabhängig, bei manchen lassen sich mehrere, verschiedene einstellen. Dann kann man seine Interrupt-Serviceroutine direkt dort ablegen und für verschiedene Peripherie-Geräte geeignete Service-Routinen direkt anspringen. Laufen alle Interrupts über nur eine Verzweigungsadresse, kann man dort eine Suchroutine unterbringen, die dann heraustüftelt, welches Peripherie-Gerät bedient werden möchte. Dann verzweigt man von dieser Suchroutine zur eigentlichen Service-Routine. Und wenn alles erledigt ist, wird das unterbrochene Programm, aus dem ohne sein Wissen verzweigt wurde, still und heimlich dort fortgesetzt, wo es verlassen wurde.

Die Methode, per externer Hardware zu einer bestimmten Adresse zu verzweigen, wenden Sie übrigens stets dann an, wenn Sie bei Ihrem Rechner einen RESET durchführen. Dabei wird allerdings nicht in ein übergeordnetes Programm zurückgekehrt, sondern nur die RESET-Adresse angesprungen und das Programm, das dort liegt, ausgeführt.

Was macht den Umgang mit Interrupts nun so knifflig? Interrupts treten im allgemeinen asynchron auf, das heißt, man weiß nicht genau wann. Darin liegt letztlich ja der Sinn der Interrupttechnik: auf zeitlich nicht exakt vorhersehbare Ereignisse umgehend zu reagieren, ohne durch Warteschleifen und dauernd eingestreute Abfragen, ob das Ereignis wohl eingetreten ist, Zeit für die Ausführung eines anderen Programmes zu vergeuden. Diese Unvorhersehbarkeit ist aber nicht unproblematisch. Da die Service-Routine irgend etwas tut (sonst gäbe es sie wohl nicht), verändert sie in bestimmter Weise auch das unterbrochene Programm. So könnte in einem Ein-/Ausgabepuffer 'plötzlich' ein Zeichen mehr oder weniger enthalten sein.

Wenn dies das unterbrochene Programm nicht mitbekommt, ergeben sich ganz sicher Probleme. In ähnlicher Weise muß man auch Sorgfalt walten lassen, wenn man (auf Maschinenebene) durch Interruptroutinen Registerinhalte verändert. Hier sollte man grundsätzlich alle Register, auf die in der Interruptroutine zugegriffen wird, vorübergehend auf dem Stack ablegen.

Und so gibt es noch eine Reihe weiterer Fehlermöglichkeiten, die die Interrupt-Technik sozusagen in Verruf gebracht haben. Aber Fehler sind schließlich dazu da, daß man sie macht, findet, behebt und daraus lernt. Und da nachweislich viele, viele Interrupt-Programme erdacht und zum Laufen gebracht wurden, kann es so schlimm nun auch wieder nicht sein. Besonders einfach wird es, wenn man über BASIC-Versionen verfügt, die die Interruptprogrammierung unterstützen.

Einfachste Lösung

In BASIC-Versionen, die das Arbeiten mit Interrupts unterstützen, kann man beispielsweise ein Programm wie in Bild 1 schreiben.

In Zeile 11 wird die Zeile 100 zur Anfangsadresse der Interruptroutine erklärt. Außerdem wird die Interruptursache vorab definiert: es soll eine Interruptanforderung gestellt werden, wenn die Taste (KEY) 3 betätigt wird. Es ist klar, daß die Schaltung des Rechners diese Möglichkeit vorsehen muß.

```
 10 REM Hauptprogramm
 11 ON KEY §3 GOSUB 100
 12 . . .
      .
      .
 22 . . .
 23 . . .
      .
      .
 99 GOTO . . .
100 REM Interruptroutine
101 PRINT "TASTE 3"
102 RETURN
103 END
```

Bild 1. Interrupts zur Tastaturabfrage in BASIC

In Bild 2 wird das Prinzip einer solchen Schaltung gezeigt. Die Tastatur wird **nicht** per Programm laufend abgefragt. Das Tastaturinterface verfügt über 'eingebaute Intelligenz' und kann per Software so eingestellt werden, daß es die Betätigung einer bestimmten Taste (hier: 3) ohne CPU-Hilfe selbst erkennt. Es meldet diese Betätigung 'hinter dem Rücken' des laufenden Programms über die Leitung $\overline{\text{INT}}$ (Interrupt-Request) direkt an die CPU.

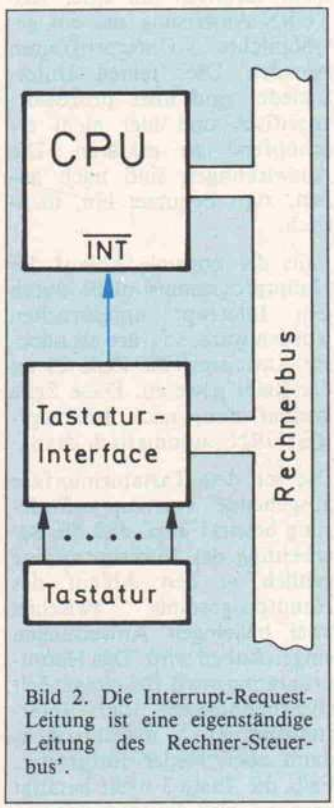


Bild 2. Die Interrupt-Request-Leitung ist eine eigenständige Leitung des Rechner-Steuerbus'.

Die Anweisung in Zeile 11 hat nur eine Initialisierungsfunktion; sie fragt die Tastatur nicht ab und bewirkt auch keine Verzweigung. Aber die Taste 3 ist ab sofort 'scharf gemacht'. Nach Bearbeitung der Zeile 11 läuft das Programm weiter in den Zeilen 12 bis 99, dem 'Hauptprogramm'. Das Hauptprogramm kann beliebig lange laufen.

Falls nun, während das Hauptprogramm läuft, die Taste 3 betätigt wird, so wird von dem Tastaturinterface durch Aktivierung der Leitung $\overline{\text{INT}}$ eine Interruptanforderung an die

CPU gestellt. Nehmen wir an, das Hauptprogramm wäre zu diesem Zeitpunkt gerade mitten in Zeile 22. Die Zeile 22 wird dann noch fertig bearbeitet werden, danach wird automatisch der Sprung (GOSUB 100) zur Interruptroutine ausgeführt.

Die Interruptroutine gibt den Text 'Taste 3' aus, danach läuft sie auf RETURN. Die RETURN-Anweisung, die eine Interruptroutine beendet, ist nicht identisch mit einer RETURN-Anweisung die ein gewöhnliches Unterprogramm beendet. Die 'feinen Unterschiede' sind aber prozessorspezifisch und hier nicht erschöpfend zu erklären. Die Auswirkungen sind nach außen, zum Benutzer hin, identisch.

Falls der normale Ablauf des Hauptprogramms nicht durch den Interrupt unterbrochen worden wäre, so wäre als nächste Anweisung die Zeile 23 an der Reihe gewesen. Diese Zeile kommt nun, nach Interrupt-RETURN, automatisch dran.

Die von dem Tastaturinterface ausgehende Interruptanforderung bewirkt also, daß die Bearbeitung der Interruptroutine **zeitlich** in den Ablauf des Hauptprogramms zwischen zwei beliebigen Anweisungen eingeschoben wird. Das Hauptprogramm wird für einige Zeit (Bearbeitungszeit der Interruptroutine) unterbrochen, dann aber wieder fortgesetzt. Falls die Taste 3 nicht betätigt wird, so wird die Interruptroutine 100 niemals durchlaufen.

Interruptquellen

Die möglichen Interruptquellen sind sehr vielfältig und hängen stark von der Rechnerhardware ab. Außer den durch Interfaces ausgelösten kennt man noch Interrupts, die durch Timer-Bausteine, Single-Step-Schaltungen, Netzausfall, Fehler im Programmablauf oder durch spezielle Programmbefehle (Softwareinterrupts) ausgelöst werden.

Prinzipiell kann jedes Interface einen Interrupt auslösen. Unter welchen Bedingungen es dies tut, wird durch seinen Typ, durch die Schaltung des Rechners und durch die softwaremäßige Initialisierung im voraus bestimmt. Ob und wann diese Bedingungen eintreten, ist, wie bereits erwähnt, nicht genau bekannt. Im Programm von Bild 1 (Tastaturinterrupt) hängt das beispielsweise vom freien Willen des Bedieners ab.

Der einfachste Einsatz von Interrupts für I/O-Zwecke läßt sich mit Timern verwirklichen.

Pseudo-I/O-Interrupts

Ein Timer (im Rechner eingebauter Zeitgeber) kann nach entsprechender Initialisierung durch ein Programm in regelmäßigen Zeitabständen einen Interrupt auslösen.

Da der Timer unabhängig von der CPU exakt wie eine Uhr läuft, weiß man, **wann** er das Hauptprogramm unterbricht. Aber **wo** innerhalb des Programms dies geschieht, bleibt trotzdem praktisch unbekannt. Um es herauszufinden, müßte man die Bearbeitungszeiten aller Anweisungen genau kennen und darüber eine Buchführung anlegen. Bei bedingten Sprüngen (IF ... THEN GOTO ...) wären dabei die unterschiedlichen möglichen Wege zu berücksichtigen. Bei Rechenoperationen müßte bedacht werden, daß sie je nach Wert der Operanden verschieden lange dauern können. Aber all dieses würde für eine Vorhersage immer noch nicht ausreichen, falls der Rechner auf Ereignisse von außen (Peripheriegeräte, Bedieneraktionen usw.) reagiert.

Der Einsatz von Timer-Interrupts ermöglicht es, trotz der eben erwähnten Ungewißheiten, bestimmte Aktionen (die in der Timer-Interruptroutine programmiert werden) in regelmäßigen Zeitabständen durchzuführen.

Im ersten Teil des Beitrags (Heft 3/84) wurde die Notwendigkeit dargelegt, bei der programmgesteuerten I/O-Technik mit Pufferung ein bestimmtes Unterprogramm (dort mit UP 2100 bezeichnet) in gewissen Zeitabständen unabhängig vom Hauptprogramm aufzurufen. Dies läßt sich mit Timer-Interrupts gut bewerkstelligen. Man legt einfach den erwähnten UP-Aufruf in eine Timer-Interruptroutine, die extra für diesen Zweck eingerichtet wird. Die Interrupts bei dieser Methode werden 'Pseudo-I/O-Interrupts' genannt. Einerseits dienen sie zu I/O-Zwecken, andererseits werden sie aber nicht durch einen I/O-Baustein, sondern durch einen Timerbaustein erzeugt.

Die entsprechende Weiterentwicklung eines Programms aus dem ersten Teil des Beitrages zeigt Bild 3. Das Prinzip der Pufferung bleibt dabei vollständig erhalten. Der Aufruf 'GOSUB 2100' im UP 2000 ist entfallen, da er ja (in Form der Timer-Interruptroutine UP 3100) sowieso laufend ausgeführt wird.

Bei allen uns bekannten BASIC-Versionen, die Interrupts in dieser Form kennen, kann die Interruptroutine nur zwischen zwei Programmzeilen durchlaufen werden. Zwischen zwei Anweisungen, die in derselben Programmzeile stehen, gibt es keine Unterbrechnung. Die Warteschleife in Zeile 3001 besteht nur aus einer einzigen Zeile. Man könnte nun denken, daß dies während des Ablaufs der Schleife den Ablauf der Interruptroutine verhindert, da es ja keine zwei Zeilen gibt, zwischen denen sie eingeschoben werden könnte. Dies ist zum Glück nicht der Fall: BASIC betrachtet den Sprung GOTO 3001 so, als führte er zu einer neuen Zeile. Diese neue Zeile ist nun zufällig wieder dieselbe Zeile.

Eine weitere Änderung findet sich in der Initialisierungsroutine UP 3300. Hier ist mit Zeile 3304 die Initialisierung für die Timer-Interruptroutine ergänzt worden: der Timer-Interrupt soll alle 35 msec erfolgen und zur Interruptroutine 3100 führen. Verwendet wird der Timer Nr. 1. Wichtig ist, daß die Timer-Initialisierung erst am Ende der gesamten Initialisierung erfolgt. Es könnte sonst passieren, daß der allererste Timer-Interrupt erfolgt, bevor noch die gesamte Initialisierung abgeschlossen ist. Die Timer-Interruptroutine greift aber sowohl auf das Interface als auch auf den Puffer zu. Sind diese beiden dann noch nicht initialisiert, so kann ein Fehler entstehen. Bei diesem Programm ist vorausgesetzt, daß der Timer sich nach Ablauf des Zeitintervalls selbst neu startet.

Ein ganz wichtiger Punkt ist, daß die Interruptroutine das Hauptprogramm überall unterbrechen kann; dies ist im Programmlisting aber nicht direkt zu sehen. Sie benutzt und verändert teilweise Variablen, die im Hauptprogramm verwendet werden. Falls man dies bei der Programmierung übersieht, kommt es zu den 'unerklärlichen' Fehlern, vor denen eingangs bereits gewarnt wurde.

Ein Beispiel:

```
200 FOR Z=1 TO 50
201 FELD(Z)=Ø
202 NEXT Z
```

Dieses Programmstück ist, für sich betrachtet, vollkommen in Ordnung. Wird es aber in einem Hauptprogramm verwendet, das die Interruptroutine 3100 (Bild 3) enthält, so kommt es zu Fehlern. Wenn die Interruptroutine zwischen den Zeilen 200 und 202 eingeschoben wird (was man nicht verhindern kann), so wird Z in unzulässiger Weise verändert. Um dies von vornherein zu vermeiden, ist es gute Programmiertechnik, in jeder Interruptroutine (und in den von ihr aufgerufenen UP's!) keine Variablen zu verwenden, die an anderer Stelle im Hauptprogramm verwendet werden. Dies sollte man selbst dann tun, wenn man genau weiß (zu wissen glaubt!?), daß kein Fehler zu befürchten ist.

Es gibt jedoch Variablen, die zwangsläufig sowohl in der Interruptroutine als auch im Hauptprogramm verwendet werden müssen. Im Beispiel von Bild 3 ist dies die Variable PUFFER$. In irgendeiner Variablen muß man ja Daten austauschen. Obwohl in diesem Fall PUFFER$ in beiden Programmteilen exakt die gleiche Bedeutung hat, gibt es trotzdem Fehlermöglichkeiten. Nehmen wir dazu an, im UP 3000 von Bild 3 hätte man Zeile 3002 in zwei Zeilen codiert:

```
3002 B$ = PUFFER$
3003 PUFFER$ = B$ + A$
```

Wenn dann die Interruptroutine zwischen diesen beiden Zeilen durchlaufen wird, kann sie den String PUFFER$ durch eine vollführte Ausgabe verkürzen. B$ bleibt davon aber unberührt, so daß in Zeile 3003 ein Fehler entsteht, der zu Zeichenverdopplungen bei der Ausgabe führt. Dieses Beispiel erscheint zwar gewaltsam konstruiert (ist es auch!), aber in der Praxis sind ja die Programme nicht immer so übersichtlich. Wenn man dann nach zwei Monaten die üblichen Programmverbesserungen vornimmt, sollte man

```
3000 REM Ausgabe A$ --> PUFFER$
3001 IF LEN(A$)+LEN(PUFFER$) > 255 THEN GOTO 3001
3002 PUFFER$ = PUFFER$ + A$
3004 RETURN


3100 REM Timer-Interrupt: PUFFER$ --> Drucker
3101 IF LEN(PUFFER$) =0 THEN GOTO 3106
3102 Z = ASC(PUFFER$)
3103 GOSUB 3200
3104 IF S=0 THEN GOTO 3106
3105 PUFFER$ = RIGHT$(PUFFER$,LEN(PUFFER$)-1)
3106 RETURN


3200 REM Drucker-Treiber
3201 S = 0
3202 IF (INP(6) AND &H81) <> &H81 THEN GOTO 3205
3203 S = 1
3204 OUT 4,Z
3205 RETURN


3300 REM Initialisierung
3301 OUT 6,0 : OUT 6,0 : OUT 6,0 : OUT 6,&H40
3302 OUT 6,&HCE : OUT 6,&H27
3303 PUFFER$ =""
3304 ON TIMER #1,35 GOSUB 3100
3305 RETURN
```

Bild 3. I/O mit Pufferung und Timer-Interrupt

```
4000 REM Ausgabe A$ --> PUFFER$
4001 IF LEN(A$)+LEN(PUFFER$) > 255 THEN GOTO 4001
4002 PUFFER$ = PUFFER$ + A$
4003 IF READY =0 THEN GOTO 4007
4004 DISABLE INTR
4005 IF LEN(PUFFER$) >0 THEN GOSUB 4300
4006 ENABLE INTR
4007 RETURN


4100 REM I/O-Interrupt: PUFFER$ --> Drucker
4101 IF LEN(PUFFER$) >0 THEN GOTO 4104
4102 READY =1
4103 GOTO 4105
4104 GOSUB 4300
4105 RETURN


4200 REM Drucker-Treiber
4201 OUT 4,Z
4202 RETURN


4300 REM Hilfs-UP: PUFFER$--> Drucker
4301 Z = ASC(PUFFER$)
4302 PUFFER$ =RIGHT$(PUFFER$,LEN(PUFFER$)-1)
4303 READY =0
4304 GOSUB 4200
4305 RETURN


4400 REM Initialisierung
4401 READY =1
4402 OUT 6,0 : OUT 6,0 : OUT 6,0 : OUT 6,&H40
4403 OUT 6,&HCE : OUT 6,&H27
4404 PUFFER$ =""
4405 ON INTR #6 GOSUB 4100
4406 RETURN
```

Bild 4. I/O-Interrupteinsatz

also zuerst wieder 'Variablen-Lernen'.

'Pseudo'-Praxis

Zunächst muß die Größe des Timer-Zeitintervalls geeignet gewählt werden. Macht man es zu klein, so wird die Interruptroutine sehr oft durchlaufen; das Hauptprogramm wird durch die häufigen Unterbrechungen stark verlangsamt. Beliebig klein läßt sich das Zeitintervall in Interpreter-BASIC ohnehin nicht machen. Zwar kann man oft den Timer in Schritten von einer Millisekunde einstellen, doch nützt das wenig. Die laufende BASIC-Zeile wird ja, wie oben beschrieben, beim Auftreten eines Interrupts zuerst fertig bearbeitet, was häufig länger als eine Millisekunde dauert. Auch wird meist die Bearbeitung der Interruptroutine selbst länger als eine Millisekunde dauern; bevor sie aber nicht zum RETURN gelangt, kann sie nicht neu aufgerufen werden.

Macht man das Zeitintervall zu groß, so wartet oft der Drucker einige Zeit, bis der nächste Interrupt kommt. Er wird also verlangsamt.

Trotz dieser Schwierigkeiten beim Einstellen des 'richtigen' Zeitintervalls ist die Pseudo-Interrupt-Technik der einfachen I/O-Technik mit Pufferung überlegen. Die Zeitintervalle sind ja in jedem Fall wegen des Pufferungsprinzips nötig und auch tatsächlich vorhanden. Bei der einfachen Technik werden sie durch Programmlaufzeiten usw. gebildet; dabei lassen sie sich schlecht festlegen und noch schlechter verändern. Außerdem können die einzelnen Intervalle sehr verschieden lang ausfallen. Das bedeutet aber, daß durch diese Schwankungen zwangsläufig von der optimalen Intervalllänge abgewichen wird.

Bei Verwendung von Timer-Interrupts läßt sich sehr einfach ein Zeitintervall mit einigermaßen genau bekannter Größe erzeugen. Änderungen zum Ausprobieren des optimalen Intervalles sind sehr einfach, da nur eine einzige Anweisung (Zeile 3304 in Bild 3) geändert werden muß. Außerdem erspart man sich das Einfügen der Aufrufe des Ausgabe-UP's überall im Programm. Da das nur zeitlich gesehen erfolgt, ohne daß tatsächlich überall die Aufrufe stehen, wird das Programm kürzer und übersichtlicher.

'Echte' I/O-Interrupts

Eine weitere Verbesserung ergibt sich, wenn man die Interrupts unmittelbar von einem I/O-Baustein auslösen läßt. Der Timer-Baustein 'weiß' ja nichts von den I/O-Vorgängen. Er unterbricht das Programm, ohne Rücksicht, präzise wie eine Uhr. Deshalb kann es nicht ausbleiben, daß die Interruptroutine manchmal durchlaufen wird, ohne daß dabei ein Zeichen ausgegeben wird; dann ist entweder der Puffer leer, oder der Drucker ist nicht 'ready'.

Die Interruptroutine wird also pro auszugebendes Zeichen im Schnitt mehr als einmal durchlaufen. Wenn man nun erreichen könnte, daß sie dafür genau einmal durchlaufen würde, so käme dabei unter dem Strich mehr verfügbare Zeit für das Hauptprogramm heraus. Diese Möglichkeit bietet sich, wenn ein Interrupt direkt von einem I/O-Baustein ausgelöst wird. Diese Auslösung erfolgt immer dann, wenn der I/O-Baustein die Notwendigkeit dazu erkennt. Da er ja mit dem entsprechenden I/O-Gerät (hier mit dem Drucker) verbunden ist, kann er es ohne CPU-Hilfe erkennen.

Ein geeigneter Zeitpunkt für die Auslösung ist der Moment, wenn der Drucker seinen Status von 'nicht ready' auf 'ready' verändert. In der Interruptroutine kann man dann sofort das nächste Zeichen ausgeben. Eine Abfrage auf 'ready' entfällt: der Interrupt wäre bei Status 'nicht ready' ja nicht erfolgt. Voraussetzung ist hierbei, daß der Status **nur** dann von 'ready' auf 'nicht ready' wechselt, nachdem ein Zeichen ausgegeben wurde, und daß dieser Wechsel auch stattfindet. Der nächste Interrupt kann erst wieder ausgelöst werden, wenn ein **Wechsel** von 'nicht ready' nach 'ready' auftritt.

Man sieht, daß die Verwendung von I/O-Interrupts genau definierte Anforderungen an die Hardware stellt. Für die Auslösung der Interrupts ist daher ein geeignetes Interface erforderlich; das I/O-Gerät muß mit seinen Handshakesignalen daran angepaßt sein, um den Interrupt im richtigen Moment auslösen zu können.

BASIC-Lösung

In BASIC kann ein Programm unter Verwendung von I/O-Interrupts wie in Bild 4 aussehen. Bei der Betrachtung des UP 4000 lassen wir zunächst die Zeilen 4003 bis 4006 außer acht. Das UP 4000 arbeitet dann wie das UP 3000 des vorigen Beispiels. Die Interruptroutine UP 4100 gibt bei jedem Durchlauf ein Zeichen aus dem Puffer aus, falls er nicht leer ist. Sie bedient sich dazu des Hilfs-UP's 4300, das ein Zeichen aus dem Puffer entnimmt und es über den Druckertreiber UP 4200 ausgibt. Der Status des Druckers wird dabei nicht mehr abgefragt: der Interrupt wird ja gerade dann ausgelöst, wenn der Drucker 'ready' wird.

Zu einer kritischen Situation kommt es, wenn die Interruptroutine einen leeren Puffer vorfindet. Dies kann sicher irgendwann auftreten, wenn das Hauptprogramm längere Zeit nichts auszugeben hat. Die Interruptroutine läuft dann auf RETURN, ohne den Drucker anzusprechen. Aber: es wird kein nachfolgender Interrupt mehr kommen! Er kommt ja nur beim **Wechsel** des Status von 'nicht ready' auf 'ready'. Nun wechselt aber der Status nicht mehr. Er war zu Beginn der Interruptroutine 'ready', und er bleibt es weiterhin, da die Interruptroutine wegen des leeren Puffers kein Zeichen ausgibt. Das Programm wird also steckenbleiben.

Selbst wenn später im Hauptprogramm durch einen Aufruf des UP 4000 wieder Daten in den Puffer geschrieben werden,

so nützt dies nichts. Es führt ja nicht zu einer Statusänderung des Druckers und damit nicht zu einem Interrupt. Ein neuer Interrupt wird erst wieder nach der Ausgabe eines Zeichens kommen. Diese Ausgabe läßt den Drucker für kurze Zeit 'nicht ready' werden. Danach wird er wieder 'ready', und ein Interrupt wird ausgelöst. Nun erfolgt diese Ausgabe aber in der Interruptroutine selbst, so daß sich hier 'die Katze in den Schwanz beißt'.

In unserem Fall ist der Ausweg aus diesem Dilemma die Einführung der Variablen READY. Sie soll anzeigen, ob der Drucker gerade ein Zeichen bearbeitet (READY = Ø), so daß ein Interrupt erwartet werden kann, oder ob der Drucker nichts zu tun hat (READY = 1). Die Interruptroutine setzt READY = 1, falls sie kein Zeichen ausgibt. Bei der Ausgabe eines Zeichens wird in jedem Fall READY = Ø gesetzt, zusammen mit dem Aufruf des Drucker-Treibers. Falls nun im UP 4000 beim Einschreiben von Daten in den Puffer READY = 1 erkannt wird, so ist kein Interrupt zu erwarten. Deshalb gibt das UP 4000 in diesem Fall ausnahmsweise selbst ein Zeichen aus dem Puffer an den Drucker aus (Zeilen 4003 bis 4006).

Es bedient sich dazu des Hilfs-UP's 4300, das auch in der Interruptroutine zur Ausgabe verwendet wird. Dadurch ist sichergestellt, daß die Ausgabe nicht in zwei unterschiedlichen Programmstücken erfolgt, die voneinander nichts wissen, und die spätestens nach der ersten Änderung nicht mehr widerspruchsfrei laufen. Nach der Ausgabe dieses einen Zeichens durch das Hauptprogramm wird wieder ein Interrupt erfolgen: die Ausgabe per Interruptroutine 'kommt wieder in Fahrt'.

Der Aufruf des UP 4300 findet sich sowohl im Hauptprogramm (Zeile 4005) als auch in der Interruptroutine (Zeile 4104) wieder. Hier lauert schon die nächste Fehlermöglichkeit. Tritt nämlich ein Interrupt auf, während das Hauptprogramm gerade das UP 4300 aufgerufen hat, so wird das UP 4300 von der Interruptroutine erneut aufgerufen, bevor es im Hauptprogramm zum Ende gekommen ist. So etwas wird als 'Rekursiv-Aufruf' bezeichnet. Je nach Programmiersprache und Codierung ist das unzulässig. Auf jeden Fall ist es sehr schwierig zu überblicken, was in einem solchen Fall genau geschieht. Deshalb wurden in Bild 4 die Zeilen 4004 und 4006 eingefügt. Mit DISABLE INTR wird in der CPU eine Interruptsperre eingeschaltet, die mit ENABLE INTR wieder aufgehoben wird. Interface und Drucker wissen von der Sperre nichts, so daß auch innerhalb der Sperrzeit eine Interruptanforderung an die CPU gestellt werden kann. Die CPU gibt dieser Anforderung aber nicht statt; sie merkt sie sich jedoch und führt den Sprung zur Interruptroutine später aus: nach Aufhebung der Sperre durch ENABLE INTR. Auf diese Weise geht also kein Interrupt 'verloren'.

Das oben geschilderte Problem fehlender Interrupts bei leerem Puffer tritt auch beim Programmbeginn auf. Der Puffer ist ja zunächst leer, der Drukker 'ready'. Das Initialisierungs-UP 4400 setzt READY = 1, damit löst sich das Problem ganz zwanglos von selbst.

Die Zeile 4405 des Initialisierungs-UP's entspricht der Zeile 3304 des vorigen Beispiels. Es soll ein Interrupt ausgelöst werden von dem Interface mit der Adresse Ø6, an das der Drucker angeschlossen ist. Dabei ist 4100 die Adresse der Interruptroutine.

Die 'Echten' in der Praxis

Wie schon aus der Beschreibung des Verfahrens zu erkennen war, ist hier noch mit erheblich mehr Fallstricken zu rechnen, als bei dem vorangegangenen Verfahren. Die bei Timer-Interrupts aufgezeigten Fehlerquellen sind hier genauso vorhanden, dazu kommen aber noch weitere. Ein Timer läuft, einmal richtig initialisiert, wie eine Uhr; solange die Interruptroutine korrekt mit RETURN abgeschlossen ist, wird sie auch immer wieder durchlaufen werden. Selbst wenn einmal ein einzelner Interrupt ausfallen sollte, oder ein überzähliger kommt, so ist das nicht tragisch.

Die Auslösung eines I/O-Interrupts erfolgt dagegen im Zusammenspiel von Gerät, Interface und Programm. Es kann erhebliche Mühe machen, sicherzustellen, daß die Interrupts stets kommen, wenn sie erwartet werden, und niemals kommen, wenn sie nicht erwartet werden. Das Verfahren steht und fällt mit der eben genannten Voraussetzung. Programmtechnische Sicherheitsmaßnahmen gegen fehlende oder überzählige Interrupts sind zwar grundsätzlich möglich, aber schwierig und ihrerseits sehr fehleranfällig.

Grundsätzlich muß man auch gerade bei der Verwendung von I/O-Interrupts mit 'hineinkonstruierten' Fehlern der Hardware, des Betriebssystems und gegebenenfalls der höheren Programmiersprache rechnen. Selbst die CPU's und I/O-Bausteine sind davon nicht ausgenommen. Bei externen Geräten (in unserem Beispiel beim Drucker) muß man auch mit gewissen Ereignissen rechnen, die zu beliebigen, vom Rechner aus unvorhersehbaren Zeitpunkten eintreten können: Ein/Aus-Schalten, Online/Offline-Umschalten, Papierende und ähnliches. Es gibt genügend Möglichkeiten, durch solche Ereignisse den Interruptmechanismus (Hardware und Software) zu stören.

Eine weitere Gruppe von Fehlern entsteht dadurch, daß das Hauptprogramm an Stellen unterbrochen wird, wo dies nicht zulässig ist. Dabei kann es sich um die oben erwähnten Rekursivaufrufe handeln. Es gibt aber auch andere Befehlsfolgen, die nicht unterbrochen werden dürfen. Grundsätzlich sind alle Anweisungsfolgen, die mit der Pufferverwaltung oder mit dem Druckerstatus zu tun haben, als gefährdet anzusehen. Allein schon das Vertauschen von zwei Anweisungen, deren Reihenfolge im Hauptprogramm eigentlich gleichgültig ist, kann im Hinblick auf mögliche Unterbrechungen unzulässig sein. Selbst wenn es im ganzen Programmlauf nur eine einzige Millisekunde gibt, in der ein Interrupt stören würde: er findet sie mit Sicherheit!

Die hier gebrachten BASIC-Programme zeigen die Fehlermöglichkeiten nicht so deutlich, sie sehen sehr simpel aus. Die Schwierigkeiten beginnen aber spätestens dann, wenn einige sinnvolle Erweiterungen (zum Beispiel der Einbau einer Formatsteuerung) oder Änderungen (beispielsweise bei der Pufferverwaltung, um von der zeitaufwendigen Stringverarbeitung loszukommen) vorgenommen werden.

Centronics-Treiber für Z 80

Nach so viel Theorie und 'Bangemacherei' nun aber etwas Praktisches:

Ein Treiberprogramm in Z 80-Assemblersprache für die Ansteuerung eines als Centronics-Schnittstelle geschalteten Z 80-PIO-Interfaces unter Benutzung von I/O-Interrupts. Das Listing ist in Bild 5 abgedruckt, die zugehörige Schaltung in Bild 6.

Die Schaltung belegt nur einen Port, und zwar Port A. Um den DATA STROBE-Impuls zu erzeugen, wird das Datenbit 7 verwendet. Es können dadurch als Daten nur die Codes $ØØ bis $7F ausgegeben wer-

```
'0000                0001 PUFEIN  EQU     $                ;1 ZEICHEN --> PUFFER
                     0002
                     0003 ;BEI AUFRUF: A = AUSGABEZEICHEN
                     0004
'0000  47            0005         LD      B,A
'0001  3AAE01'       0006 PUFEI1  LD      A,(PUFBLZ)       ;BELEGUNGSZAEHLER
'0004  FEFF          0007         CP      255              ;VOLL ?
'0006  28F9          0008         JR      Z,PUFEI1-$       ;JA --> ERST LEEREN
'0008  2AAF01'       0009         LD      HL,(PFEINZ)      ;EINSCHREIBEZEIGER
'000B  70            0010         LD      (HL),B           ;ZEICHEN --> PUFFER
'000C  23            0011         INC     HL               ;EINSCHREIBEZEIGER +1
'000D  EB            0012         EX      DE,HL
'000E  21AE01'       0013         LD      HL,PUFBLZ
'0011  34            0014         INC     (HL)             ;BELEGUNGSZAEHLER +1
'0012  21AE01'       0015         LD      HL,PUFEND+1
'0015  B7            0016         OR      A
'0016  ED52          0017         SBC     HL,DE            ;PUFFERENDE UEBERSCHRITTEN ?
'0018  2003          0018         JR      NZ,PUFEI2-$      ;NEIN
'001A  11AF00'       0019         LD      DE,PUFANF        ;JA --> ZURUECK ZUM ANFANG
'001D  ED53AF01'     0020 PUFEI2  LD      (PFEINZ),DE      ;NEUER EINSCHREIBEZEIGER
'0021  3A8301'       0021         LD      A,(READY)
'0024  B7            0022         OR      A                ;INTERRUPT ERWARTET ?
'0025  F3            0023         DI
'0026  C42B00'       0024         CALL    NZ,PUFAUS        ;FALLS NEIN --> ANSTOSSEN
```

Fortsetzung Bild 5 nächste Seite

```
'0029  FB        0025          EI
'002A  C9        0026          RET
                 0027
'>002B           0028 PUFAUS   EQU     $               ;PUFFER --> DRUCKER
                 0029
                 0030 ;BEIM RUECKSPRUNG: FLAG NONZERO, FALLS PUFFER NICHT LEER WAR
                 0031 ;                  FLAG ZERO, FALLS LEER
                 0032
'002B  3A0E01'   0033          LD      A,(PUFBLZ)      ;BELEGUNGSZAEHLER
'002E  B7        0034          OR      A               ;LEER ?
'002F  2821      0035          JR      Z,PUFAU2-$      ;JA
'0031  3D        0036          DEC     A
'0032  32AE01'   0037          LD      (PUFBLZ),A      ;NEIN --> -1
'0035  2A8101'   0038          LD      HL,(PFAUSZ)     ;AUSLESEZEIGER
'0038  7E        0039          LD      A,(HL)          ;LFD. AUSGABEZEICHEN
'0039  CD6E00'   0040          CALL    DTREIB          ;DRUCKER-TREIBER AUFRUFEN
'003C  23        0041          INC     HL              ;AUSLESEZEIGER +1
'003D  EB        0042          EX      DE,HL
'003E  21AE01'   0043          LD      HL,PUFEND+1
'0041  B7        0044          OR      A
'0042  ED52      0045          SBC     HL,DE           ;PUFFERENDE UEBERSCHRITTEN ?
'0044  2003      0046          JR      NZ,PUFAU1-$     ;NEIN
'0046  11AF00'   0047          LD      DE,PUFANF       ;JA --> ZURUECK ZUM ANFANG
'0049  ED53B101' 0048 PUFAU1   LD      (PFAUSZ),DE     ;NEUER AUSLESEZEIGER
'004D  AF        0049          XOR     A
'004E  32B301'   0050          LD      (READY),A       ;READY =0 NACH AUSGABE
'0051  3C        0051          INC     A               ;--> FLAG NONZERO
'0052  C9        0052 PUFAU2   RET
'>0053           0053 INTRPT   EQU     $               ;INTERRUPT-ROUTINE
'0053  F5        0054          PUSH    AF              ;ALLE VERWENDETEN REGIS-
'0054  C5        0055          PUSH    BC              ;TER RETTEN
'0055  D5        0056          PUSH    DE
'0056  E5        0057          PUSH    HL
'0057  3AB301'   0058          LD      A,(READY)
'005A  B7        0059          OR      A
'005B  200A      0060          JR      NZ,INTRP1-$     ;DUERFTE NIE KOMMEN
'005D  CD2B00'   0061          CALL    PUFAUS          ;AUSGABE AUS PUFFER
'0060  2005      0062          JR      NZ,INTRP1-$     ;PUFFER WAR NICHT LEER !
'0062  3E01      0063          LD      A,1
'0064  32B301'   0064          LD      (READY),A       ;PUFFER LEER --> READY =1
'0067  E1        0065 INTRP1   POP     HL              ;REGISTER WIEDERHERSTELLEN
'0068  D1        0066          POP     DE
'0069  C1        0067          POP     BC
'006A  F1        0068          POP     AF
'006B  FB        0069          EI
'006C  ED4D      0070          RETI
                 0071
                 0072
                 0073
'>006E           0074 DTREIB   EQU     $               ;DRUCKER-TREIBER
                 0075
                 0076 ;BEI AUFRUF: A = AUSGABEZEICHEN
                 0077
'006E  2F        0078          CPL                     ;WEIL 7404 INVERTIERT
'006F  CBFF      0079          SET     7,A
'0071  D312      0080          OUT     (12H),A         ;DATEN + STROBE HIGH
'0073  CBBF      0081          RES     7,A
'0075  D312      0082          OUT     (12H),A         ;DATEN + STROBE LOW
'0077  CBFF      0083          SET     7,A
'0079  D312      0084          OUT     (12H),A         ;DATEN + STROBE HIGH
'007B  C9        0085          RET
                 0086
                 0087
                 0088
'>007C           0089 INIT     EQU     $               ;INITIALISIERUNG
'007C  F3        0090          DI
'007D  AF        0091          XOR     A
'007E  32AE01'   0092          LD      (PUFBLZ),A      ;PUFFER = LEER
'0081  3C        0093          INC     A
'0082  32B301'   0094          LD      (READY),A       ;READY =1
'0085  21AF00'   0095          LD      HL,PUFANF       ;PUFFERZEIGER INITIALISIEREN
'0088  22AF01'   0096          LD      (PFEINZ),HL
'008B  22B101'   0097          LD      (PFAUSZ),HL
'008E  ED5E      0098          IM      2
'0090  21B501'   0099          LD      HL,INTVEK+1     ;ADRESSE DES INTERRUPTVEKTORS
'0093  CB85      0100          RES     0,L             ;MUSS GERADE SEIN !!!!
'0095  7D        0101          LD      A,L
'0096  D314      0102          OUT     (14H),A         ;LOW BYTE --> PIO
'0098  7C        0103          LD      A,H
'0099  ED47      0104          LD      I,A             ;HIGH BYTE --> I-REGISTER
'009B  115300'   0105          LD      DE,INTRPT       ;ADRESSE DER INTERRUPT-ROUTINE
'009E  73        0106          LD      (HL),E          ;EINTRAGEN IN IN-
'009F  23        0107          INC     HL              ;TER-
'00A0  72        0108          LD      (HL),D          ;RUPTVEKTOR
'00A1  3E80      0109          LD      A,10000000B
'00A3  D312      0110          OUT     (12H),A         ;STROBE HIGH
'00A5  3E8F      0111          LD      A,00001111B
'00A7  D314      0112          OUT     (14H),A         ;PIO-MODE =0 (OUTPUT)
'00A9  3E97      0113          LD      A,10010111B
'00AB  D314      0114          OUT     (14H),A         ;ENABLE PIO-INTERRUPT
'00AD  FB        0115          EI
'00AE  C9        0116          RET
                 0117
                 0118 ;ARBEITSSPEICHER:
                 0119
                 0120 PUFANF   DEFS    255             ;RINGPUFFER
'>01AD           0121 PUFEND   EQU     PUFANF+254      ;ENDADRESSE DES PUFFERS
                 0122
                 0123 PUFBLZ   DEFS    1               ;BELEGUNGSZAEHLER
                 0124 PFEINZ   DEFS    2               ;EINSCHREIBEZEIGER
                 0125 PFAUSZ   DEFS    2               ;AUSLESEZEIGER
                 0126
                 0127 READY    DEFS    1               ;MERKER FUER DRUCKER-STATUS
                 0128 INTVEK   DEFS    3               ;INTERRUPTVEKTOR AUF GERA-
                 0129 ;                                ;DER ADRESSE
                 0130          END

ERRORS=0000
```

Bild 5. Interruptgesteuerter Centronics-Treiber

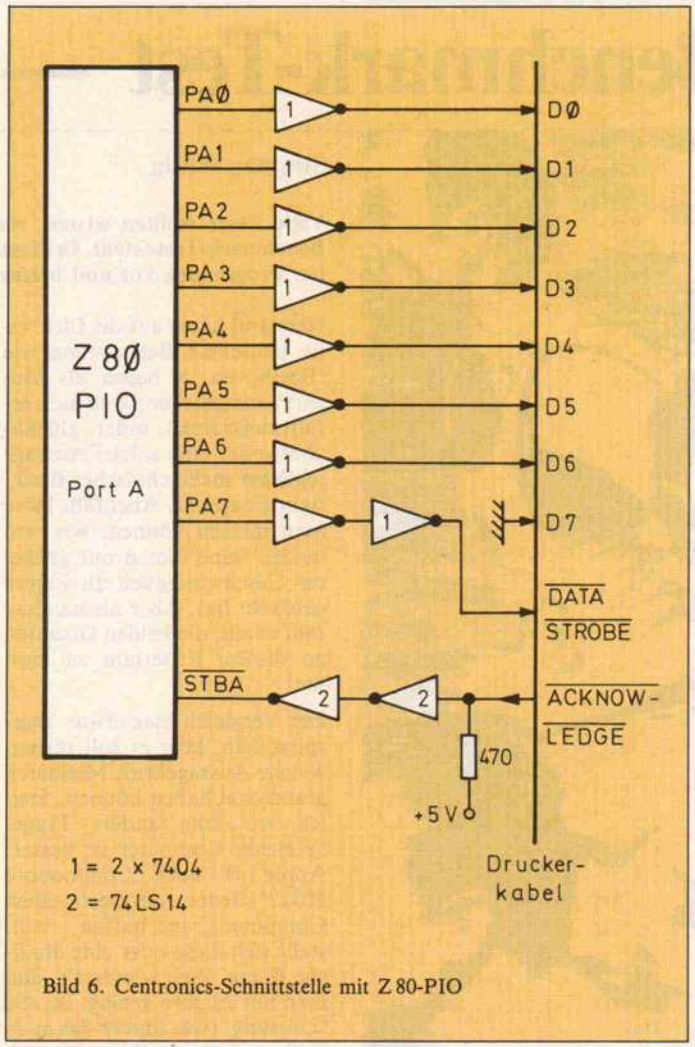


Bild 6. Centronics-Schnittstelle mit Z80-PIO

den. Die PIO wird per Software so eingestellt, daß der ACKNOWLEDGE-Impuls des Druckers den Interrupt auslöst. Die BUSY-Leitung ist bei diesem Verfahren überflüssig.

Das Assemblerprogramm in Bild 5 besteht aus 5 UP's. Es wird auch hier der schon in der ersten Folge des Beitrags beschriebene Ringpuffer verwendet. PUFEIN dient im Hauptprogramm zur Ausgabe eines Zeichens in den Ringpuffer, PUFAUS ist ein Hilfs-UP zur Entnahme eines Zeichens aus dem Puffer und dessen Ausgabe an den Drucker-Treiber DTREIB. Letzterer erzeugt den DATA STROBE-Impuls mit einer Folge von 3 Ausgabebefehlen. Die Interruptroutine INTRPT wird durch den ACKNOWLEDGE-Impuls des Druckers ausgelöst. Das UP INIT erledigt die hier sehr vielfältigen Aufgaben beim Programmstart. Dabei schaltet es auf Interrupt-Mode 2, belegt einen Platz für den Interruptvektor im Bereich dieses Programms und lädt das I-Register entsprechend. Dies kann nur funktionieren, wenn (mit Ausnahme von NMI) keine anderen Interrupts im Rechner auftreten. Falls ein anderer Programmteil oder das Betriebssystem mit Interrupts arbeitet, so entsteht durch das UP INIT ein Fehler: entweder beim Umschalten des Interrupt-Modes oder beim Umladen des I-Registers. Wird an mehreren Stellen im Programm (Betriebssystem eingeschlossen) mit Interrupts gearbeitet, so müssen diese Aktionen koordiniert werden. Es muß überall der gleiche Interrupt-Mode verwendet werden; bei Mode 2 müssen alle Interruptvektoren in einer gemeinsamen 'Vektortabelle' liegen, auf die dann das I-Register in Verbindung mit den Lower-Byte-Interruptvektoren der Portbausteine zeigt. □

Benchmark-Test — eine Computer-Meßl

Siegmar Wittig

Viele Leser wollten wissen, welche 'Philosophie' hinter den c't-Benchmark-Tests steht. In diesem Bericht stellen wir die verwendeten Programme vor und beleuchten die Hintergründe.

Niemand käme auf die Idee, eine pauschale Behauptung wie 'Beethoven ist besser als Mozart' aufzustellen, weil sich erfahrungsgemäß (oder glücklicherweise) eine solche Aussage jeglicher meßtechnischer Bestätigung entzieht. Allenfalls hätte man messen können, wer von beiden seine Noten mit größerer Geschwindigkeit zu Papier gebracht hat. Aber niemandem fällt es ein, die beiden Giganten an diesem Kriterium zu messen!

Der Vergleich mag etwas überspitzt sein, aber er soll zeigen, welche Aussagekraft Meßdaten manchmal haben können. Stellen wir eine andere Frage: 'Welcher Computer ist besser, Apple II oder Commodore 8032?' Jeder, der sich einen Computer anschaffen will, stellt sich diese oder eine ähnliche Frage. Wen wundert's, daß man nur zu gern geneigt ist, die 'Leistung' (was immer das auch heißen mag) von Computern in Zahlen (wenn möglich mit etlichen Stellen hinter dem Komma) zu 'messen'.

Der Wunsch(traum) ist so alt wie die Computer selbst. Einige der folgenden Bewertungsverfahren spielen in der Praxis eine gewisse Rolle. Werfen wir einen Blick darauf.

Was ist ein Mix?

Ein 'Mix' ist ein Bewertungsverfahren, bei dem man die Ausführungszeit für eine typische 'Mischung' (daher Mix) von Einzeloperationen bestimmt. Solche Mixes werden für bestimmte Problembereiche (zum Beispiel kommerzielle Anwendungen) eigens zusammengestellt. Hierbei erhalten die einzelnen Operationen unterschiedliche Gewichte, je nach Anwendungsgebiet, für das der Computer (genauer: die Zentraleinheit) vorgesehen ist. 'Mix I' ist zum Beispiel der Name eines solchen Mixes für kommerzielle Anwendungen. Vergleichsbefehle haben hier ein relativ hohes Gewicht, Rechenoperationen spielen für die Bestimmung der 'Kennzahl' nur eine untergeordnete Rolle.

Der 'GIBSON-Mix' dagegen soll ein Maß für die Geschwindigkeit bei der Ausführung von mathematischen und technischen Problemstellungen liefern. Dieser Mix ist nicht einheitlich definiert, aber man kann soviel sagen, daß Rechenoperationen hier natürlich ein wesentlich höheres Gewicht haben als andere Operationen.

Für mathematisch-technische Anwendungsprogramme war auch der 'GAMM-Mix' gedacht (GAMM ist die Abkürzung für 'Gesellschaft für Angewandte Mathematik und Mechanik'). Dieser Mix besteht aus einer Reihe typischer mathematischer Operationen. Hier zwei Beispiele: Aus 100 Zahlen soll die Zahl mit dem größten Betrag gesucht werden. Oder: Berechnung einer Quadratwurzel. Solche und andere mathematische Aufgaben hat der Computer zu lösen. Die Zeiten hierfür werden gewichtet und ergeben zusammen die gesuchte Mixkennzahl.

Gemischte Gefühle

Mixes der eben genannten Art sind und waren immer sehr umstritten. Denn beispielsweise wird die Rechengenauigkeit, eine Eigenschaft des Rechners, die die Ausführungszeiten wesentlich mitbestimmt, nicht berücksichtigt. Dürfte es doch auf der Hand liegen, daß zwei ansonsten gleiche Computer bei einer Rechengenauigkeit von sechs beziehungsweise zehn signifikanten Stellen unterschiedliche Ausführungszeiten für mathematische Operationen haben müssen!

Benchmark-Tests

Kaum ein Computer-Testbericht ohne Benchmark-Test! Benchmark-Tests sind, wie die Mixes, ein Versuch, die Leistungsfähigkeit einer Datenverarbeitungsanlage zu bewerten (wir vermeiden das Wort 'messen'). Ein Benchmark-Test wird mit einer Reihe ausgesuch-

Foto: Hanns Krebs (Bavaria)

```
100 REM BM1
300 PRINT "S"
400 FOR K=1 TO 1000
500 NEXT K
700 PRINT"E"
800 END
```

Programm 1. Zählschleife. Die Variable K nimmt der Reihe nach die Werte von 1 bis 1000 an.

```
100 REM BM2
300 PRINT"S"
400 K=0
500 K=K+1
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
```

Programm 2. Zählschleife wie in Programm 1, jedoch realisiert mit einer IF-THEN-Anweisung.

```
100 REM BM3
300 PRINT"S"
400 K=0
500 K=K+1
510 A=K/K*K+K—K
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
```

Programm 3. Der Zählschleife aus Programm 2 wurden arithmetische Operationen mit der Variablen K zugefügt.

```
100 REM BM4
300 PRINT"S"
400 K=0
500 K=K+1
510 A=K/2*3+4—5
600 IF K<1000 THEN 500
700 PRINT"E"
```

Programm 4. Das gleiche Programm wie Nr. 3, jedoch werden hier die arithmetischen Operationen mit Konstanten ausgeführt. Wegen der dazu notwendigen Zahlenkonversionen wird Programm 4 langsamer als Programm 3 sein.

```
100 REM BM5
300 PRINT"S"
400 K=0
500 K=K+1
510 A=K/2*3+4—5
520 GOSUB 820
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
820 RETURN
```

Programm 5. Wie Programm 4, jedoch mit zusätzlichen Sprüngen in ein 'leeres' Unterprogramm.

```
100 REM BM6
300 PRINT"S"
400 K=0
430 DIM M(5)
500 K=K+1
510 A=K/2*3+4—5
520 GOSUB 820
530 FOR L=1 TO 5
540 NEXT L
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
820 RETURN
```

Programm 6. Wie Programm 5, jedoch zusätzlich mit der dimensionierten Variablen M(.) und einer kurzen geschachtelten FOR-NEXT-Schleife (Zeilen 530 bis 540).

```
100 REM BM7
300 PRINT"S"
400 K=0
430 DIM M(5)
500 K=K+1
510 A=K/2*3+4—5
520 GOSUB 820
530 FOR L=1 TO 5
535 M(L)=A
540 NEXT L
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
820 RETURN
```

Programm 7. Wie Programm 6, jedoch zusätzlich mit 5000 Zuweisungen zu einer Bereichsvariablen (Zeile 535).

```
100 REM BM8
300 PRINT"S"
400 K=0
500 K=K+1
530 A=K↑2
540 B=LOG(K)
550 C=SIN(K)
600 IF K<100 THEN 500
700 PRINT"E"
800 END
```

Programm 8. Dieses Programm soll die Ausführungszeit für transzendente Funktionen messen. Hier zeigen BASIC-Interpreter oft erhebliche Zeitunterschiede.

ter Programme durchgeführt, die für den Benutzer typische Aufgaben lösen sollen. Ein Benchmark-Test ist also immer eine individuelle Angelegenheit. Der Benutzer sucht mit einem solchen Test nach dem Computer, der seine Programme am schnellsten ausführt.

Als Tom Rugg und Phil Feldman 1977 in der US-amerikanischen Zeitschrift Kilobaud zwei Artikel über den Vergleich der Ausführungszeiten von Mikrocomputern schrieben (Heft 6, Seite 66 bis 70, und Heft 10, Seite 20 bis 25), ahnten sie nicht, welche Konsequenzen ihre dort veröffentlichten 'Benchmarks' haben sollten. Ihre sieben kurzen Benchmark-Programme (siehe Kasten) sind nachgerade zu einem Standard für Mikrocomputer-Tests geworden. Dabei ist hier die Bezeichnung 'Benchmark' äußerst unglücklich, wenn nicht sogar unpassend gewählt. Bei genauem Hinsehen handelt es sich nämlich eher um Teilprogramme einer Art von Mix, denn in den kurzen Programmen werden spezielle BASIC-Anweisungen in Kombination von Rechenoperationen getestet, aber nicht problemorientierte, benutzerspezifische Anwendungen. Aber die Bezeichnung 'Benchmark' hat sich nun mal durchgesetzt, was uns einmal mehr resignierend die vielbeschworene 'normative Kraft des Faktischen' zu zitieren Anlaß gibt.

Fast alle Computertests in c't enthielten eine Messung der Ausführungszeiten der sieben 'Benchmark'-Programme. Hinzu kam noch ein achtes Programm, das von John A. Coll in der Zeitschrift Computer Education, Heft 28, Seite 18—21, veröffentlicht wurde (siehe Kasten). Auf Wunsch zahlreicher Leser drucken wir diese acht Programme hier noch einmal ab. In den zurückliegenden Heften von c't können Sie anhand der dort veröffentlichten Ausführungszeiten der acht Programme ihre eigene 'Maschine' einordnen.

Damit wir uns richtig verstehen: Auch die c't-Tester wissen, daß Mixes und Benchmark-Tests einen beschränkten Wert für die Beurteilung eines Computers haben. Wir sagten bereits, daß bei solchen Vergleichen überhaupt nicht berücksichtigt wird, mit welcher Genauigkeit der Computer Dezimalzahlen verarbeitet. Und über die Leistungsfähigkeit der Peripheriegeräte wird schon gar nichts ausgesagt, von Benutzerfreundlichkeit, Zuverlässigkeit und Qualität der Dokumentation ganz zu schweigen. Zum Messen von Ausführungszeiten braucht man halt nur eine Stoppuhr (oder nicht einmal diese, wenn der Computer eine eingebaute Uhr hat), die anderen Merkmale aber sind nur mit Hilfe langer Checklisten zu bestimmen — oder gar nicht. Das ist das Geheimnis der unverdienten Beliebtheit von Mixes und Benchmarks! □

anpassen

Teil 2: Kommunikation mit FORTH

Peter Glasmacher

Der erste Teil dieser Serie beschäftigte sich mit den Basisvereinbarungen, die nötig sind, um FORTH auf einem Tischcomputer lauffähig zu bekommen. Am Beispiel des APPLE II wurde aufgezeigt, welche Überlegungen v o r Eingabe des Quellcodes notwendig sind, um allzu aufwendige Entlausungsprozeduren zu vermeiden. Im vorliegenden Teil 2 soll FORTH nun die Fähigkeit bekommen, über Ein-/Ausgabeeinheiten mit dem jeweiligen Herrn und Meister zu kommunizieren.

FORTH gehört zu den interaktiven Sprachen, deshalb sollte der Computer mit einer alphanumerischen Tastatur und einem Videodisplay ausgerüstet sein, so wie es zum Beispiel beim Apple der Fall ist. Des weiteren sollte das Betriebssystem des Rechners in der Lage sein, ein Zeichen von der Tastatur zu übernehmen und auf dem Bildschirm auszugeben. Diese beiden Grundfunktionen, Terminalein- und -ausgabe werden von FORTH mit zwei Worten, **KEY** und **EMIT** abgewickelt. EMIT sendet das oberste Byte auf dem FORTH-Stack zu einem Terminal und KEY übernimmt **ein** Zeichen von der Tastatur und legt es auf den Stack. Zwei weitere Worte unterstützen die Basisfunktionen:

CR sendet einen Wagenrücklauf zur Ausgabeeinheit und ?TERMINAL fragt den Eingabekanal ab und legt ein logisches 'Wahr' oder 'Falsch' auf den Stack, abhängig davon, ob zum Zeitpunkt der Abfrage eine Taste gedrückt war oder nicht. Aus diesen Basisworten wird anschließend in bester FORTH-Manier ein ganzes Paket von Ein-/Ausgabefunktionen gebildet, eine Zusammenfassung zeigt Tabelle 1.

Nun ist es durchaus möglich, mit dem im FIG-Listing abgedruckten KEY und EMIT Code-Fragmenten (Bild 1) die gesamte Ausgabe zu bestreiten. Mehr als Basisoperationen sind jedoch nicht möglich. Es wäre wünschenswert, vom Programm die verschiedenen Ausgabegeräte wie Drucker und 80-Zeichenkarte zu berücksichtigen und außerdem der Tatsache Rechnung zu tragen, daß dem APPLE II einige Tasten fehlen. So ist zum Beispiel die eckige Klammer (Ä) nicht auf der Tastatur zu finden.

Wenn man der Familie jedoch schon die Zeit für ein Projekt wie eine FORTH-Implementation abgetrotzt hat, sollte man sich die spätere Arbeit auch so bequem wie möglich machen und von vornherein die verschiedenen 'Ausgänge' mit berücksichtigen.

Programmierter Komfort

Gerade der APPLE ist das typische Beispiel einer 'Featuremaschine'. Es gibt wohl kaum einen Anwender, der seinen APPLE ohne irgendwelche Zusatzkarten betreibt. In der Regel sind zumindest eine 80-Zeichen-Karte und eine Druckerschnittstelle in einen der 8 Erweiterungsstecker (SLOT) gesteckt — womit die Probleme beginnen. Das liegt nicht an der nackten Existenz, sondern an der Tatsache, daß jede Karte auf eine andere Weise initialisiert und über verschiedene Steuerzeichen bedient und kontrolliert wird. Die meisten 80-Zeichen-Karten lassen sich zum Beispiel nicht mit dem APPLE-Befehl 'HOME' löschen, sondern verlangen dafür meist ein CTRL-L. Printerkarten benötigen oft eine Initialisierungssequenz, die nur über die Eingabe von 'PR#' aufgerufen werden kann. Ein anderer Weg wäre, ein an einen Drucker auszugebendes Zeichen direkt in das entsprechende Register zu laden und das ebenfalls vorhandene Strobe-Register solange abzufragen, bis der Drucker wieder bereit ist. Der letzte Vorschlag ist für FORTH (fast) ideal. Wird solch eine direkte Druckausgabe in EMIT eingebaut, kann das gesamte Durckergeschäft über nur ein Flag gesteuert werden.

Für die korrekte Ausführung von 'HOME' hat der Verfasser ein FORTH-Wort 'HOME' definiert und zusätzlich in EMIT eine Abfrage eingebaut, welche die gerade aktuelle Ausgabeeinheit feststellt und das korrekte 'HOME' ausführt. Zuletzt sollte dann noch die CR-Routine für alle Ausgabeeinheiten angepaßt werden (Bild 2), da CR

TABELLE 1. FORTH-TERMINAL I/O

KEY	(--- c)	Ein Zeichen einlesen.
EMIT	(c ---)	Ein Zeichen ausgeben.
?TERMINAL	(--- f)	Wahr wenn Taste gedrueckt.
CR	(---)	Sendet CR zum Terminal.
EXPECT	(addr n ---)	Liest n Zeichen und legt sie unter addr ab.
TYPE	(addr u ---)	Sendet die Zeichenkette in der Laenge u (max 65k !) unter der Adresse addr.
.R	(n m ---)	Sendet die Zahl n rechtsbuendig in einem m Zeichen grossen Feld.
WORD	(c ---)	Liest die direkt auf WORD folgenden Zeichen ein und stellt die Zeichenkette an die NAECHSTE FREIE STELLE in die Bibliothek. c ist ein Trennzeichen.

Tabelle 1. Zusammenfassung der Ein-/Ausgabefunktionen

```
                492 *
                493 *     'XKEY' SCR=7, L=4
                494 *
7511:20 0C FD   495 XKEY      JSR INCH          ; ZEICHEN EINLESEN
7514:29 7F      496           AND #$7F          ; BIT 7 AUSBLENDEN
7516:4C DC 64   497           JMP PUSH0A        ; -->TOS
                498 *
                499 *     'XEMIT' SCR=7, L=8
                500 *
7519:B5 00      501 XEMIT     LDA $00,X         ; ZEICHEN VON TOS
751B:09 80      502           ORA #$80          ; BIT 7 SETZEN
751D:20 ED FD   503           JSR OUTCH         ; ZEICHEN SENDEN
7520:4C EE 62   504           JMP POP           ; VOM STACK ENTF.
                505 *
                506 *     'XCR' SCR=7, L=10
                507 *
7523:20 8E FD   508 XCR       JSR TCR           ; CR ZUM BILDSCHIRM
7526:4C 44 61   509           JMP NEXT          ; UND WEITER
```

Bild 1. Die Basisroutinen XKEY, XEMIT und XCR. Außer Tastatur und Bildschirm wird nichts unterstützt.

MTX-512: DER PROFI ZUM HOBBYPREIS

Schon wieder ein Neuer mögen Sie klagen und am liebsten umblättern. Aber an der MTX-SERIE kommen Sie nicht ungestraft vorbei. Technologisch von der Mittleren Datentechnik stammend, bietet die MTX-SERIE eine Vielzahl von Eigenschaften, die anderswo erst mit Software und Peripherie erkauft werden muß. So läßt sich schon das Grundgerät für die verschiedensten Spezialanwendungen einsetzen.

KAUFLEUTE werden die große Tastatur, den deutschen Zeichensatz, das numerische Tastenfeld, den Monitorausgang und den Centronics-Druckerausgang ebenso zu schätzen wissen, wie die Option, das System mit Diskettenstation unter CP/M® zu fahren (wem dann die maximale Speicherkapazität von 32 Megabyte Pseudofloppy und 20 Megabyte Winchester nicht genügt: bitte, dann müssen wir passen).

INGENIEUREN und Technikern dagegen imponiert da schon eher die eingebaute Echtzeituhr, der parallele I/O, der zugängliche Systembus, die umfassende technische Dokumentation sowie der im ROM implementierte ASSEMBLER/DISASSEMBLER (die Option mit 512 K RAM interessiert da schon weniger).

FREIZEITSPIELER, aber nicht nur diese, sondern auch Musiker und die eher auf Grafik erpichten Hobbyisten werden entzückt sein über die LOGO-gesteuerte Grafik, 32 Sprites mit 7 Parametern und dem 3-Kanal-HiFi-Tonausgang unter Softwaresteuerung (16 Farben und 2 Joystick Ports helfen natürlich auch weiter).

PROGRAMMIERER – professionell oder „just for fun" – werden die Gesamtheit aller Eigenschaften zu würdigen wissen, denn sie sind meist die Leidtragenden einer mangelhaft durchkonzipierten Maschine. Wer mit dem Befehl PANEL sofort alle Registerzustände, den Inhalt des RAMs und die Mnemonics des anstehenden Programms erfährt, weiß was er hat. Wer in einer BASIC-Zeile ASS. NNN eingibt, um anschließend eine ASSEMBLER-Routine zu schreiben, um dann in der Zeile NNN+1 in BASIC fortzufahren, ist schon ein Besserwisser (daß Sie dank des getrennten 16 K Bildschirmspeichers und dem selekt. 24 K ROM volle 64 K als Arbeitsspeicher zur Verfügung haben, soll hier nicht weiter stören).

ANFÄNGER, aber auch terminge-mierer können Memotech für das im ROM vorhandene NODDY danken. Mit nur sieben Befehlen lassen sich dialogorientierte Menüs, aber z.B. auch Bildschirmmasken ohne PRINT-Anweisung erstellen – kinderleicht (weshalb NODDY in England für den Anfangsunterricht eingesetzt werden soll).

ALLE sollten wissen, daß die Memotech-MTX-SERIE in 2 Grundversionen erhältlich ist (MTX-500/32 K RAM zu DM 1198,–; MTX-512/64 K RAM zu DM 1390,–), von Profisoft, dem „Software-Haus mit Service" vertrieben wird und Sie gegen Zusendung einer Postkarte weitere technische Unterlagen erhalten können. Selbstverständlich haben wir auch für den Fachhandel Unterlagen – mit anderem Schwerpunkt!

Memotech MTX-512 mit Diskettenstation

MEMOTECH MTX SERIES

profisoft

Sutthauser Str. 50–52 · 4500 Osnabrück · Tel. 05 41/5 39 05

```
E561:63 E5      648 HOME    DW  *+2             ; CODE
E563:AC CF D7   649         LDY SPNT            ; POINTER AUF SFLAG
E566:B1 E3      650         LDA (UP),Y          ; SFLAG LADEN
E568:D0 06      651         BNE C80             ; <>0= 80-ZEICHEN KARTE
E56A:20 5B FC   652         JSR $FC5B           ; CALL MONITOR HOME
E56D:4C 69 D0   653         JMP NEXT            ; UND WEITER
E570:A9 0C      654 C80     LDA #$0C            ; VIDEOTERM HOME
E572:CA         655         DEX
E573:CA         656         DEX                 ; PLATZ AUF STACK
E574:94 01      657         STY 1,X             ; LUECKENFUELLER
E576:95 00      658         STA 0,X             ; $0C-->TOP
E578:D0 1E      659         BNE XEMIT           ; IMMER
E57A:           660 *
E57A:           661 * *** 'XKEY', SCR=7, L=4
E57A:           662 *
E57A:           663 XKEY    JSR INCH            ; ZEICHEN EINLESEN
E57D:29 7F      664         AND #$7F            ; BIT 7 MASKIEREN
E57F:20 85 E5   665         JSR CCHECK          ; TEST AUF SONDERZ.
E582:4C A5 D3   666         JMP PUSH0A          ; AB NACH FORTH !
E585:           667 *
E585:           668 * TEST AUF ^O,^K,^L
E585:           669 *
E585:C9 0F      670 CCHECK  CMP #$0F
E587:D0 02      671         BNE NC1
E589:A9 5F      672         LDA #$5F            ; '_'
E58B:C9 0C      673 NC1     CMP #$0C
E58D:D0 02      674         BNE NC2
E58F:A9 5C      675         LDA #$5C
E591:C9 0B      676 NC2     CMP #$0B
E593:D0 02      677         BNE NC3
E595:A9 5B      678         LDA #$5B
E597:60         679 NC3     RTS
E598:           680 *
E598:           681 * *** 'XEMIT', SCR=7, L=6
E598:           682 *
E598:86 E5      683 XEMIT   STX XSAVE           ; POINTER RETTEN
E59A:B5 00      684         LDA 0,X             ; ZEICHEN HOLEN
E59C:2C 3A 0C   685 XEMIT1  BIT PFLAG           ; AUSGABE AUF DRUCKER ?
E59F:10 03      686         BPL VOUT            ; NEIN, NUR VIDEO
E5A1:20 BE E5   687         JSR POUT            ; ZEICHEN  --> DRUCKER
E5A4:09 80      688 VOUT    ORA #$80            ; BIT 7=ON
E5A6:20 ED FD   689         JSR OUTCH           ; UEBERGABE MONITOR
E5A9:A6 E5      690         LDX XSAVE           ; POINTER LADEN
E5AB:4C 7B D1   691         JMP POP             ; ZURUECK
E5AE:           692 *
E5AE:           693 * *** 'XCR', SCR=7, L=8
E5AE:           694 *
E5AE:2C 3A 0C   695 XCR     BIT PFLAG           ; DRUCKER ?
E5B1:10 05      696         BPL VCR             ; VIDEO
E5B3:A9 0D      697         LDA #$0D            ; CR
E5B5:20 C0 E5   698         JSR WLOOP           ; AUSGABE DRUCKER
E5B8:20 8E FD   699 VCR     JSR TCR             ; UEBERGABE MONITOR
E5BB:4C 69 D0   700         JMP NEXT            ; ZURUECK
```

Bild 2. Die geänderten Routinen reagieren nun auf eine 80-Zeichen-Karte und die etwas abweichende Apple-Tastatur. Ein Drucker wird ebenfalls unterstützt.

ebenfalls in der FIG-Originalversion nur auf den Bildschirm wirkt.

Der Vollständigkeit halber soll auch noch eine andere Art der flexiblen Terminal I/O angesprochen werden. Einige FORTH-Systeme, zum Beispiel das FORTH des AIM 65, greifen auf KEY und EMIT über eine USER-Variable (UKEY und UEMIT) zu, in die zur Kalt- oder Warmstartzeit die Code Feld Adresse (CFA) der jeweiligen Standarddefinition geladen wird. In KEY bzw. EMIT ist dann nur noch die Phrase UEMIT (UKEY) & EXECUTE enthalten. Der Inhalt der Variablen kann dann bei Bedarf durch die CFA einer anwenderspezifischen Routine ersetzt werden. Diese Art der I/O-Steuerung ist natürlich unter Umständen etwas gefährlich und sollte mit Vorsicht gehandhabt werden.

Die Änderungen ... konkret

Bild 2 zeigt eine Zusammenfassung der geänderten Terminal-I/O-Routinen für die APPLE-Familie. In diesen Worten werden auch die im ersten Teil des Artikels vereinbarten Parameter SFLAG und PFLAG (in BOOT2, erinnern Sie sich?) verwendet. Direkt im ersten als CODE-Fragment ausgeführten Wort HOME in Bild 2 wird als erstes ein Zeiger in das Y-Register geladen, welcher die neue USER-Variable SFLAG adressiert und fragt, ob diese ungleich Ø ist. In diesem Fall wird **nicht** die Monitorroutine des Apple angesprungen, die den Bildschirm löscht und den Cursor in die Anfangsposition bringt, sondern der Wert $ØC auf den (FORTH) Stack gelegt und nach EMIT verzweigt. $ØC entspricht aber einem Drucker-FORMFEED-Kommando oder eben einem HOME für die meisten 80-Zeichen-Karten. Kenner werden bemerken, daß mit der Anweisung STY 1,X keine Null als höherwertiges Byte auf den Stack gelegt wird. In diesem Fall ist dies jedoch ohne Bedeutung, da EMIT nur das niederwertige Byte sendet und der Stack danach gelöscht wird.

Die Erweiterung von KEY ist als eigenes Unterprogramm ausgeführt, um eventuelle spätere Erweiterungen zu erleichtern. In der Routine CCHECK wird nun abgeprüft, ob eine der Control-Tasten gedrückt wurde, die als Ersatz für die fehlenden Zeichen Ö, _ und Ä herhalten müssen. CCHECK wurde vom Autor als einfache Abfrageroutine ausgeführt. Dabei wurde auf mögliche Programmiertricks verzichtet, obwohl sicher einige Bytes einzusparen wären, da die unteren Nibbles der 'fehlenden' Tasten mit denen der Control-Tasten korrespondieren, und man die Zeichen demnach durch ein 'ORA #$80' an der richtigen Stelle bilden kann.

Die beiden Ausgaberoutinen **CR** und **EMIT** wurden ebenfalls nicht verschont und präsentieren sich erheblich verändert.

CR prüft als erstes PFLAG und sendet ein $ØD zum Drucker, falls Bit 7 in PFLAG gesetzt ist. Ist PFLAG jedoch Ø, wird nur die Monitorroutine TCR aufgerufen. Dies ist bei einer Druckerausgabe, wie sie hier verwendet wird, nötig, wie später noch klar wird.

Um die Ausgabe auch auf einen Drucker zu lenken, wurde eine Druckausgabe in FORTH eingebunden. Die 65XX Prozessorfamilie kennt keine speziellen Ein-/Ausgabeanweisungen, sondern betrachet einen I/O-Kanal als eine normale Speicherstelle, in die Werte gespeichert und aus der Werte geladen werden. Um nun ein Zeichen zu einem Drucker zu bringen, benötigt man ein Ausgaberegister für das auszugebende Zeichen und ein Eingaberegister, in dem der aktuelle Druckerstatus, 'bereit' oder 'nicht bereit', gespeichert wird. Eine Ausgaberoutine muß nun dieses Status oder Stroberegister solange testen, bis der Drucker 'bereit' signalisiert (WLOOP). Danach kann das Zeichen in das Ausgaberegister geschrieben und an den Drucker übergeben werden.

Die in POUT angegebenen Adressen $C090 und $C1C1 beziehen sich auf eine der 'Standard'-Centronics-Interfacekarten für den APPLE und müssen eventuell geändert werden. Ebenso wäre es möglich, einen Einsprung in eine Ausgaberoutine für eine serielle Schnittstelle einzubinden.

Die erste Tätigkeit von POUT ist ein weiterer Test von PFLAG. Bei gesetztem Bit 6 in PFLAG wird zum Label PTAG verzweigt, in dem hier nur ein Rücksprung programmiert ist. Was auf den ersten Blick wie ein überflüssiges Verschwenden von Maschinenzeit aussieht, hat seinen Grund. Manche Drucker scheren sich nicht um den ASCII-Code und legen verschiedene, meist oft benötigte Zeichen, auf unkonventionelle Plätze. Sollte man einen dieser Drucker besitzen, kann man unter dem Label PTAB eine eigene Umwandlungstabelle samt Laderoutine ablegen und nach Durchlaufen dieser Routine seinem Drucker den richtigen Code für ein bestimmtes Zeichen anbieten.

Sind alle diese Schritte durchlaufen, springt das Programm nach EMIT zurück und gibt das Zeichen noch auf dem Bildschirm aus; die nötigen Befehle mit einer angedeuteten Tabelle zeigt Bild 3.

(Wird fortgesetzt)

```
E5BE:           701 *
E5BE:           702 * *** PRINT UNTERPROGRAMM
E5BE:           703 *
E5BE:70 09      704 POUT    BVS PTAB            ; USER DEFINED CGEN?
E5C0:2C C1 C1   705 WLOOP   BIT $C1C1           ; STROBE  ?
E5C3:30 FB      706         BMI WLOOP           ; NEIN, ZURUECK
E5C5:8D 90 C0   707         STA $C090           ; AUSGEBEN !!!
E5C8:60         708         RTS
E5C9:           709 *
E5C9:           710 * *** PTAB IST EIN VORBEREITETER SPRUNG
E5C9:           711 * *** UM EVENTUELL IN EINER EIGENEN ROUTINE
E5C9:           712 * *** SPEZIELLE DRUCKERZEICHEN UMZUWANDELN
E5C9:           713 *
E5C9:4C C0 E5   714 PTAB    JMP WLOOP           ; NICHT IMPLEMENTIERT

 1 PTAB  TAX                ; ZEICHEN ALS INDEX BENUTZEN
 2       LDA TAB,X          ; ALTERNATIVES ZEICHEN LADEN
 3       JMP WLOOP          ; AUSGEBEN
 4 *
 5 TAB   HEX 000102030405060708  ; UND SO WEITER
```

Bild 3. Auf diese Weise kann jedes Zeichen durch ein anderes ersetzt werden, natürlich wird eine komplette Tabelle benötigt.

mcs

NEU! NEU! NEU! NEU! NEU! NEU! NEU! NEU! NEU!

16/32-K-RAM- + 8-K-ROM-Supermodul für den VC-20.
Das ist die preiswerteste Lösung für mehr Speicher im VC-20. Ersetzt 3 K + 8 K + 16 K oder 8 K + 16 K + 8 K im Autostartbereich. Außerdem ist noch ein EPROM-Steckplatz für 4 oder 8 K EPROM vorhanden.
Preis mit 16 K RAM bestückt ..... **DM 128.–**
Preis mit 32 K RAM bestückt ..... **DM 178.–**
2 RAMs zum Umbau von 16 auf 32 K ..... **DM 60.–**

64-K-RAM-Supermodul.
Die beste aller Speichererweiterungen für den VC-20. Ersetzt alle anderen Speichermodule von 3 K bis 32 K. Erweitert den Speicher des VC-20 um volle 64 KByte, wenn mindestens ein 8-K-Bereich im VC-20 frei ist. Wer noch mehr Speicher braucht, kann bis zu vier dieser Module gleichzeitig verwenden und erhält so über 260 KByte RAM. Für die Bereiche BLK3 und BLK5 ist softwareseitig ein Schreibschutz schaltbar.
Preis für das 64-K-Supermodul ..... **DM 278.–**

80-Zeichen-Supermodul für VC-20.
Völlig neues Modul mit vielen zusätzlichen Funktionen. Das kleinste Modul dieser Art. Es paßt in ein Standard-Modulgehäuse und damit auch in die Modulbox. Auf diesem Modul befindet sich außerdem noch ein komplettes TEXTPROGRAMM im EPROM.
Preis für das 80-Z.-Supermodul ..... **DM 278.–**

Z-80A[1]-Supermodul für den VC-20.
Macht aus Ihrem VC-20 einen vollwertigen Z-80-Rechner mit 4-MHz-Z-80A-CPU, 64 KByte RAM und DMA-Interface zum VC-20 sowie wahlweise dazu einen Floppydisk-Controller zum Anschluß von 5,25"-Diskettenlaufwerken. Das Diskettenformat ist TRS-80[2]-(Double Density)kompatibel. Mit Rechnerbetriebssystem und CP/M[3]-BIOS im ROM. 60 K freier Speicher für CP/M.
Preis für das Z-80A-Supermodul ..... **DM 398.–**

Floppydisk-Controller für 1–3 Laufwerke, Single oder Double Side, 40/80 Track, Double Density, Anschlußkabel für 3 Diskettenlaufwerke ..... **DM 348.–**
Achtung! Das Z-80A-Modul erfordert eine 80-Zeichenkarte für eine einwandfreie Funktion der mitgelieferten Software!

Achtung! An alle Bastler und Freaks. Wegen Umstellung der Produktion verkaufen wir eine größere Anzahl von Platinen unserer alten VC-20-Module, z. B. Busplatine mit 6 Steckplätzen 30 DM, 80-Z.-Karte, vollbestückt (mit kleinen Fehlern) 90.– DM, sowie diverse andere Module.
Für den kleinen Geldbeutel haben wir außerdem noch diverse Gebrauchtgeräte vorrätig, z. B. Drucker ab 400.– DM, mehrere Video-Genie-Computer ab 600.– DM.

[1] Z-80A ist ein eingetragenes Warenzeichen der Firma Zilog Inc.
[2] TRS-80 ist ein eingetragenes Warenzeichen der Firma Tandy Corp.
[3] CP/M ist ein eingetragenes Warenzeichen der Firma Digital Research.

Die absolute Sensation!!!
Alle unsere VC-20-Module laufen jetzt auch am COMMODORE 64!!!

Busadapter für ein VC-20-Modul ..... **DM 48.–**
Busadapter für drei VC-20-Module ..... **DM 148.–**

Bus-Supermodul für den VC-20.
Busplatine mit fünf Steckplätzen für Module, 3-K-RAM-Erweiterung und einem Steckplatz für 4/8 K EPROM. Eigene Stromversorgung, Selectleitungen über Schalter geführt.
Preis für das Bus-Supermodul ..... **DM 228.–**

Software für Commodore VC-20:

Super-Speed Graphic Toolkit.
Das einzige Grafikprogramm für den VC-20, mit dem Sie in Basic bewegte Grafiken erzeugen können.
Erfordert mindestens ein 64-K-Supermodul.
Preis für das Graphic Toolkit ..... **DM 99.–**

Pseudo-Disk-Dateiprogramm.
Programmierkomfort wie nie zuvor. Laden und Abspeichern von Programmen und Daten in Bruchteilen von Sekunden. Arbeitet mit Basicbefehlen (LOAD, SAVE, PRINT#, INPUT#, GET#), die jedoch teilweise erweitert wurden (APPEND von Programmen und RANDOM-Datei). Erfordert mindestens ein 64-K-Supermodul.
Preis für Pseudo-Disk ..... **DM 49.–**

40 Zeichen/Zeile und Grafik.
Erlaubt die Darstellung von 40 Zeichen und 25 Zeilen mit dem VC-20 ohne Extra-Hardware. Dazu kommen Grafikbefehle zur Darstellung beliebiger Grafiken ..... **DM 49.–**

Peripheriegeräte für alle Computer:

12"-Monitor BMC BM12.
18 MHz, grüne oder bernsteinfarbene Bildröhre.
Preis für BM12 ES (grün) ..... **DM 368.–**
Preis für BM12 EY (bernsteinfarben) ..... **DM 398.–**
Sonderangebot: BM 12A ..... **DM 288.–**

Matrixdrucker BMC PX-80.
80 Zeichen pro Sekunde, Superschriftbild (9×13-Matrix), 40 bis 142 Zeichen/Zeile, Subscript und Superscript, Kursivschrift, Fettdruck, Unterstreichen und Grafik. Friktionswalze und verstellbare Traktorführung.
Preis mit Centronics-Interface ..... **DM 998.–**
Preis mit VC-20-/C-64-Interface ..... **DM 1148.–**

Tintenstrahldrucker,
superleise, mit 1a Schriftbild. 80 Zeichen/Sekunde. Serienmäßig mit Centronics- + V-24-Interface.
Preis für diesen Superdrucker ..... **DM 1548.–**
Preis mit VC-20-/C-64-Interface ..... **DM 1698.–**

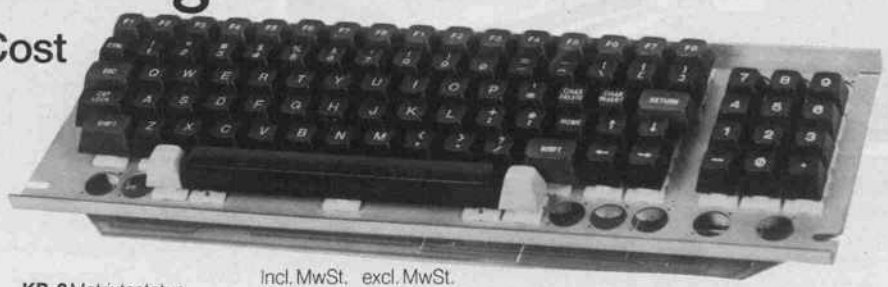
In allen Preisen ist die gesetzliche Mehrwertsteuer enthalten. Für alle Lieferungen gelten ausschließlich unsere allgemeinen Liefer- und Zahlungsbedingungen.

MicroComputerSysteme Ingeborg Strie
Kirchweg 5, 2831 Schwaförden, Tel. 04277/692

Auch als Bausatz

PROTON *intelligente* Tastaturen

High Quality - Low Cost



PROTON-Tastaturen sind mit Keyswitches von **Futuba,** weltführender Hersteller, aufgebaut. Diese Keyswitches werden auch von führenden Terminal-Herstellern wie **Lear-Siegler** und **Televideo** eingesetzt.

PROTON-Tastaturen werden in Holland hergestellt. Neben den standardmäßigen Tastaturen sind auch kundenspezifische Tastaturen preislich sehr attraktiv, auch in kleineren Stückzahlen.

Befestigungsplatte aus Stahlblech. Auf die Platine wird keine mechanische Kraft ausgeübt: Ein zuverlässiges Funktionieren ist damit gewährleistet.

ASCII-Encoder mit wählbarer Tastenbelegung und **parallelem** und **seriellem** ASCII-Ausgang mit wählbaren Schittstellen-Daten. Größte Flexibilität: Anpassung an jeden Computer möglich.

16 programmierbare Funktiontasten. Unter jede der Funktiontasten können Sie einen String bis zu 15 Charactern in EPROM ablegen, so daß Änderungen - auch nachträglich - problemlos sind. Von der Tastatur aus können diese Strings vorübergehend überschrieben werden (in das interne RAM). Standardmäßig sind die am häufigst benutzten BASIC-Befehle abgelegt.

		Incl. MwSt.	excl. MwSt.
KB-2 Matrixtastatur 10 x 10	DM	239.40	210.00
KB2E-G komplett gebaute Tastatur mit Encoder und Gehäuse.	DM	416.10	365.00
KB2E-B wie KB2E-G, jedoch als Bausatz	DM	324.90	285.00
QWERTZ Umbausatz auf deutsche Tatastenbelegung incl. EPROM.	DM	27.82	24.40
APPLECABLE Flachbandkabel (1 m) mit zwei 16-pol. IC-Steckern für Apple-Anschluß	DM	38.76	34.00

Bitte fordern Sie sofort das Informationsmaterial an!
OEM's fragen Sie gezielt an!

PROTON

In den Preisen sind 14% MwSt. enthalten. Soweit nichts anderes vereinbart, erfolgt der Versand gegen Nachnahme. Pauschalbetrag für Versand und Verpackung 9,50 DM.

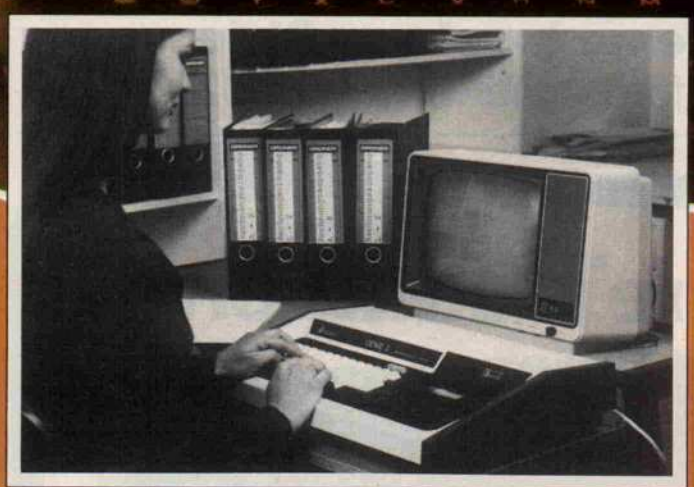
TEEPE GmbH
Vorm Tor 8 / D-6395 Weilrod
Telefon 06083/2329/553

Software-Know-how

Genie als ASCII-Fernschreiber

Diethelm Buttkus

Als Anregung für erste Versuche auf dem Gebiet der Datenfernübertragung möge dieser Beitrag dienen. Zugleich bietet er eine Anleitung zum Erstellen von Programmen für den seriellen Datenverkehr in Eintonverfahren. Wer einen Genie-Computer besitzt, kann die Software unverändert abtippen. Besitzern anderer Z80-Systeme wird die Anpassung durch eine ausführliche Beschreibung erleichtert.



In jedem Computer sind alle wichtigen Baugruppen einer elektronischen Fernschreibmaschine bereits enthalten. Es ist daher nicht schwierig, ein geeignetes Programm zu entwickeln, das über den meist vorhandenen Kassettenanschluß das Senden und Empfangen von Fernschreibzeichen gestattet.

Beim Fernschreiben erfolgt die Datenübertragung seriell und asynchron im Start-Stopp-Betrieb. Sender und Empfänger sind nur während der Übermittlung eines Zeichens annähernd synchron. Danach wartet der Empfänger beliebig lange in Ruhestellung, bis das nächste Zeichen eintrifft.

Aus geschichtlichen Gründen werden die Zeichen beim Amateur-Fernschreiben und zum Teil auch noch bei kommerziellen Fernschreibsendungen nach dem CCITT-Nr. 2-Code verschlüsselt, der auf Jean Maurice Baudot (1845—1903), einen französischen Telegrafentechniker, zurückgeht. Dies ist ein zweiwertiger Code (ein — aus ≙ Mark — Space) mit fünf Signalelementen, der stets mit einem Startschritt (Space) beginnt und einem Stoppschritt von mindestens eineinhalb Mark-Längen endet.

Auch im Computer wird jedem Zeichen ein Code zugeordnet, der aber vom CCITT-Code abweicht. Es ist der ASCII-Code, ebenfalls zweiwertig, aber mit sieben Signalelementen, denen stets eine Null (Space) vorangestellt ist.

Gegenüber dem CCITT-Code hat der ASCII-Code den Vorteil, daß er mehr Kombinationsmöglichkeiten besitzt. Es ist nicht mehr nötig, zwischen einer Buchstaben- und Zahlenebene zu unterscheiden, auch können mehr verschiedene Zeichen sowie Groß- und Kleinbuchstaben gesendet werden.

Soll ein Computer-Fernschreiber mit dem CCITT-Code arbeiten, sind mehrere Tabellen und Umschaltroutinen erforderlich, um den vorhandenen ASCII-Code in den CCITT-Code umzusetzen. Bleibt man dagegen im ASCII-Code, so gestaltet sich das Programm recht einfach.

Ausstattung

Im folgenden wird ein kleines Programm beschrieben, mit dem sich Fernschreibversuche durchführen lassen. Es sind dazu zwei Computer erforderlich, die über eine Fernleitung und ein einfaches Interface an ihren Kassettenschnittstellen verbunden sind. Für die erste Erprobung genügt es, die Zeichen auf einem Kassettenrecorder aufzunehmen, der dann anschließend als Sender benutzt wird. Ein Interface ist solange noch nicht nötig.

Leider läuft das Programm nur auf GENIE-Computern unverändert, bei anderen Systemen müssen die Ein- Ausgabeadressen geändert werden. Auch der sonst kompatible TRS-80 hat eine andere IN/OUT Organisation.

Wird die 'F1'-Taste am Computer gedrückt, so kann man den eingebauten Recorder verwenden. Eingriffe in das Gerät sind nicht erforderlich. Die Übertragung erfolgt im ASCII-Code, nach dem Eintonverfahren. Bei 'Mark' liegen auf der Fernleitung 0 Volt, bei 'Space' ein Ton, oder besser ein 'Burst', mit einer Frequenz von 1275 Hz. Die Baudrate, das heißt die Anzahl der pro Sekunde übertragenen Bits, ist zwischen 45.45 Baud und 100 Baud wählbar. Die Übertragung ist nur wenig störanfällig, so daß auch die akustische Einkopplung auf eine Fernleitung oder drahtlose Übermittlung möglich sind. Dabei sind die postalischen Vorschriften zu beachten!

Die gesendeten oder empfangenen Zeichen erscheinen nach jedem Tastendruck gleichzeitig auf dem Bildschirm beider Geräte, wobei auch Korrekturen mit der Linkspfeiltaste möglich sind. Beim üblichen Fernschreiben wäre das undenkbar. Wird ein eventuell vorhandener Drucker eingeschaltet, so wird der Text auch ausgedruckt.

Software

Das Programm (Tabelle 1) besteht aus zwei Teilen, einem Maschinenprogramm und einem kleinen BASIC-Programm, das die Baudrate einstellt. Nach dem Laden beider

Programmteile erfolgt die Aufforderung, die gewünschte Baudrate einzugeben. Das BASIC-Programm berechnet nun alle Verzögerungszeiten und setzt mit 'POKE' alle Zeitschleifen auf den richtigen Wert, sofern die gewünschte Baudrate von der Grundeinstellung 45.45 Baud abweicht. Danach erscheint in der rechten unteren Bildschirmecke ein 'E'. Der Computer zeigt damit an, daß das Gerät empfangsbereit ist. Mit der NEW LINE Taste kann man auf Sendung umschalten. Dies wird durch ein 'S' in der rechten unteren Bildschirmecke signalisiert.

Betrachten wir kurz den Programmablauf, und beginnen wir mit 'Senden'.

Die Tastatur wird abgefragt und das geforderte Zeichen auf Bildschirm und gegebenenfalls Drucker ausgegeben. Mit der BREAK-Taste kann die Sendung unterbrochen werden, und das Gerät schaltet wieder auf Empfang. Wird der 'Klammeraffe' gedrückt, kehrt der Computer in das BASIC-Programm zurück, etwa um die Baudrate zu ändern.

Das Zeichen wird nun in das E-Register geladen. Dann 'testet' der Prozessor den Wert jedes einzelnen Datenbits. Nach jeder Prüfung wird der Inhalt des E-Registers um ein Bit verschoben, bis alle acht Bit erkannt sind. Danach springt das Programm zur Tastaturabfrage zurück, um ein neues Zeichen zu übernehmen.

Wurde ein Bit als Null erkannt (Space), erfolgt ein Sprung zum SPACE-Unterprogramm. Die Datenleitungen D0 und D1 werden abwechselnd HIGH und LOW geschaltet, so daß am Ausgang von Port 'FF' eine Rechteckschwingung mit einer Frequenz von 1275 Hz und einer Dauer von 22 ms (bei 45.45 Baud) anliegt. Wird ein Bit als Eins erkannt (Mark), so geschieht im MARK-Unterprogramm 22 ms lang nichts (bei 45.45 Baud).

Das Signal wird leider einem internen Spannungsteiler auf etwa 0,4 $Volt_{ss}$ abgeschwächt. Es liegt zwischen den Pins 2 und 5 an der Kassettenausgangsbuchse. Bevor es auf die Fernleitung gegeben wird, muß es in einem kleinen Interface wieder verstärkt werden, wenn man keinen Eingriff in den Computer vornehmen will. Einen Vorschlag für ein entsprechendes Interface finden Sie in Bild 1.

Die Empfangsroutine verläuft in wesentlichen Teilen umgekehrt. Zunächst wird das Eingangs-Flip-Flop von Port 'FF' zurückgesetzt. Hierzu wird ein kleines Unterprogramm (CALL 021Eh) aus dem ROM bemüht. Danach wartet der Prozessor in einer Schleife auf den Startschritt. Dieser ist bei ASCII genau wie bei CCITT eine Null, also ein Burst. Dieser Burst triggert das Flip-Flop, und die Schleife wird verlassen.

Nun prüft der Prozessor in der Mitte der Übertragungszeit für jedes Bit den Status des Flip-Flops. Es folgt daher eine Verzögerungsschleife, welche die weitere Abfrage auf die Schrittmitte verschiebt. Nun wird während einer kurzen Abtastzeit darauf gewartet, ob das Flip-Flop getriggert wird (Spa-

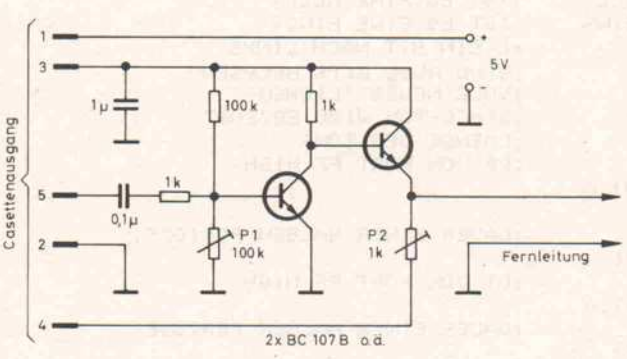


Bild 1. Interface für die Fernleitung

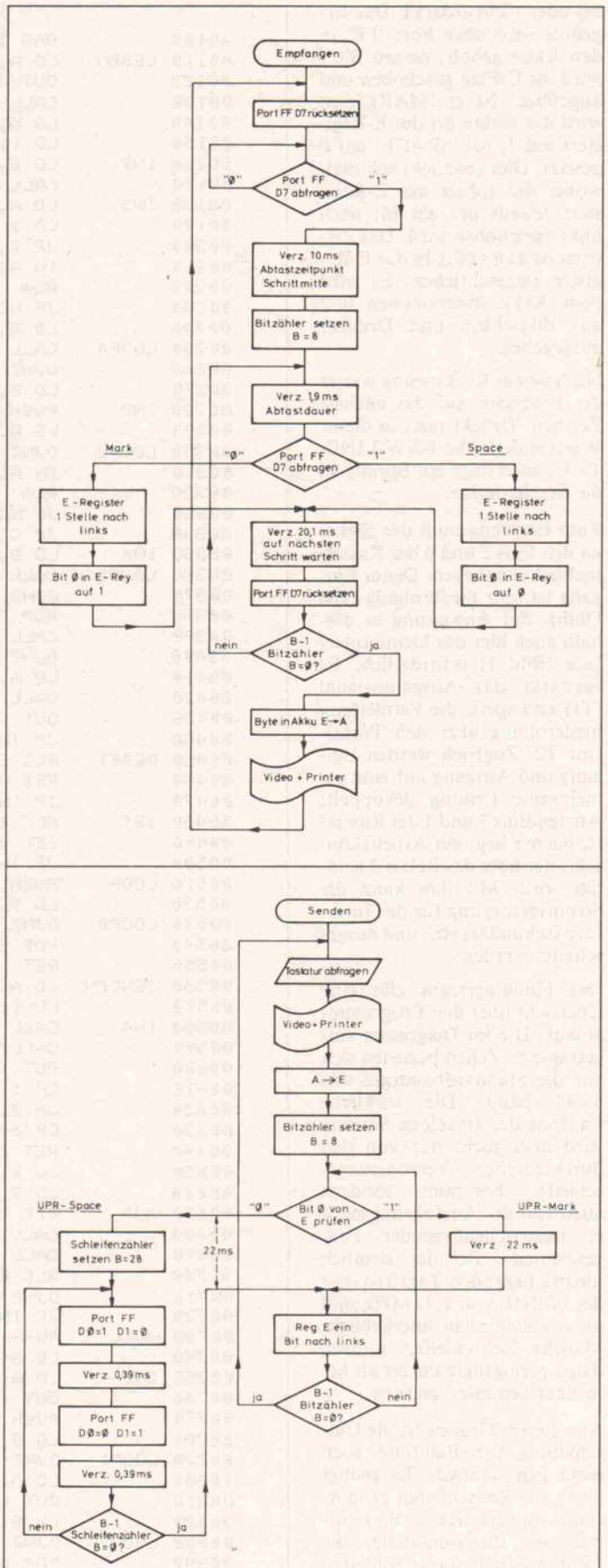


Bild 2. Flußdiagramm zu Tabelle 1

ce) oder nicht (Mark). Das Ergebnis wird über Port 'FF' in den Akku geholt, dessen Bit 7 wird ins C-Flag geschoben und abgefragt. Ist es 'MARK', so wird das rechte Bit des E-Registers auf 1, bei 'SPACE' auf 0 gesetzt. Dies geschieht achtmal, wobei der Inhalt des E-Registers jeweils um ein Bit nach links verschoben wird. Das Zeichen ist nun völlig in das E-Register eingeschrieben. Es wird vom Akku übernommen und auf Bildschirm und Drucker ausgegeben.

Nach einem Rücksprung wartet der Prozessor auf das nächste Zeichen. Drückt man in dieser Warteschleife die NEW-LINE-Taste, so erfolgt ein Sprung in die Senderoutine.

Zum Empfang muß das Signal an den Pins 2 und 4 der Kassettenbuchse anliegen. Dieser Eingang ist sehr niederohmig (100 Ohm). Zur Anpassung ist deshalb auch hier das kleine Interface (Bild 1) erforderlich. Es verstärkt das Ausgangssignal (T1) und speist die Fernleitung niederohmig über den Transistor T2. Zugleich werden Eingang und Ausgang auf eine gemeinsame Leitung gekoppelt. An den Pins 3 und 1 der Kassettenbuchse liegt ein Arbeitskontakt, der über das Relais 2 betätigt wird. Mit ihm kann die Stromversorgung für das Interface (sekundär) ein- und ausgeschaltet werden.

Das Flußdiagramm gibt eine Übersicht über den Programmablauf. Die im Diagramm eingetragenen Zeilen beziehen sich auf die Standardbaudrate von 45.45 Baud. Die wirkliche Laufzeit der einzelnen Schritte wird aber nicht nur von den durchlaufenen Verzögerungsschleifen bestimmt, sondern auch von der Ausführungsdauer dazwischenliegender Programmteile. Bei der ziemlich niedrig liegenden Taktfrequenz des GENIE von 1,77 MHz sind diese Zeiten nicht unerheblich. Manche Zeitschleifen müssen daher geringfügig kürzer als berechnet bemessen werden.

Aus diesem Grunde ist die Umschaltung der Baudrate auch nicht ganz einfach. Es genügt nicht, die Zeitschleifen proportional zu verkürzen. Die Laufzeit der Programmteile, die nicht innerhalb dieser Schleifen liegen, bleibt ja unverändert erhalten. Daher müssen die Zähler etwas überproportional ver-

```
00100           ORG 32000
00110 LESEN     LD A,16          ;SELECT KASSETTE 2
00120           OUT (0FEH),A
00130           CALL 215H        ;CASSETTE AN
00140           LD A,69          ;E AUF BILDSCHIRM UNTEN RECHTS
00150           LD (16383),A
00160 IN9       LD E,0FFH
00170           CALL 021EH       ;EINGANGS-FLIP-FLOP RUECKSETZEN
00180 IN5       LD A,(14400)     ;PRUEFEN OB NEW LINE GEDRUECKT
00190           CP 1
00200           JP Z,SENDEN      ;WENN JA, DANN AUF SENDEN SCHALTEN
00210           IN A,(0FFH)      ;AUF STARTSCHRITT WARTEN
00220           RLA              ;BIT 7 VON AKKU IN C-FLAG
00230           JP NC,IN5
00240           LD B,25          ;ABTASTZEITPUNKT EINSTELLEN
00250 LOOP4     CALL LOOP
00260           DJNZ LOOP4
00270           LD B,8           ;BIT-ZAEHLER SETZEN
00280 IN8       PUSH BC
00290           LD B,255         ;ABTASTZEIT EINSTELLEN
00300 LOOP3     DJNZ LOOP3
00310           IN A,(0FFH)      ;BIT VON PORT FF HOLEN
00320           RLA              ;BIT 7 VON AKKU IN C-FLAG
00330           JP NC,SET        ;IST ES MARK?
00340           JP C,RESET       ;IST ES SPACE
00350 IN6       LD B,40          ;AUF NAECHSTEN SCHRITT WARTEN
00360 LOOP5     CALL LOOP
00370           DJNZ LOOP5
00380           POP BC
00390           CALL 021EH       ;FLIP-FLOP RUECKSETZEN
00400           DJNZ IN8         ;SIND ALLE 8 BIT IN E?
00410           LD A,E           ;BYTE IN AKKU
00420           CALL 033H        ;ZEICHEN AUF BILDSCHIRM
00430           OUT (0FDH),A     ;ZEICHEN AUF DRUCKER
00440           JP IN9           ;HOLE NEUES ZEICHEN
00450 RESET     RLC E            ;E EINE STELLE NACH LINKS
00460           RES 0,E          ;RECHTES BIT AUF 0
00470           JP IN6           ;HOLE NEUES BIT
00480 SET       RLC E            ;E EINE STELLE NACH LINKS
00490           SET 0,E          ;RECHTES BIT AUF 1
00500           JP IN6           ;HOLE NEUES BIT
00510 LOOP      PUSH BC          ;GEMEINSAME INNERE VERZ.SCHLEIFE
00520           LD B,62
00530 LOOP0     DJNZ LOOP0
00540           POP BC
00550           RET
00560 SENDEN    LD A,83          ;S AUF BILDSCHIRM UNTEN RECHTS
00570           LD (16383),A
00580 IN4       CALL 049H        ;TASTATURABFRAGE
00590           CALL 033H        ;ZEICHEN AUF BILDSCHIRM
00600           OUT (0FDH),A     ;ZEICHEN AUF DRUCKER
00610           CP 1             ; IST ES BREAK?
00620           JP Z,LESEN       ;WENN JA, DANN AUF EMPFANG
00630           CP 64            ;IST ES KLAMMERAFFE?
00640           RET Z            ;WENN JA, DANN ZURUECK IN BASIC
00650           LD E,A           ;ZEICHEN VON AKKU IN E-REGISTER
00660           LD B,8           ;BIT-ZAEHLER SETZEN
00670 IN3       BIT 7,E          ;LINKES BIT IN E PRUEFEN
00680           CALL Z,NULL      ;IST ES EINE NULL?
00690           CALL NZ,EINS     ;IST ES EINE EINS?
00700           RLC E            ;E EIN BIT NACH LINKS
00710           DJNZ IN3         ;SIND ALLE BITS GELESEN?
00720           JP IN4           ;HOLE NEUES ZEICHEN
00730 NULL      PUSH BC          ;SPACE-TON WIRD ERZEUGT
00740           LD B,28          ;LAENGE DES TONS
00750 IN1       LD A,5           ;D0 VON PORT FF HIGH
00760           OUT (0FFH),A
00770           PUSH BC
00780           LD B,50          ;DAUER EINER HALBEN PERIODE
00790 LOOP1     DJNZ LOOP1
00800           LD A,6           ;D1 VON PORT FF HIGH
00810           OUT (0FFH),A
00820           LD B,50          ;DAUER EINER HALBEN PERIODE
00830 LOOP2     DJNZ LOOP2
00840           POP BC
00850           DJNZ IN1         ;IST SPACE ZU ENDE?
```

Tabelle 1. Unterprogramme für Empfangen (Lesen) und Senden

```
00860           POP BC
00870           RET
00880 EINS      PUSH BC          ;LAENGE VON MARK ABWARTEN
00890           LD B,46          ;OHNE TON
00900 LOOP6     CALL LOOP
00910           DJNZ LOOP6
00920           POP BC
00930           RET              ;ZURUECK WENN MARK-PAUSE ZU ENDE
00940           END
```

```
10 CLS:PRINT"BITTE WARTEN!"
20 DATA 62,16,211,254,205,21,2,62,69,50,255,63,30,255,205,30
30 DATA 2,58,64,56,254,1,202,97,125,219,255,23,210,17,125,6
40 DATA 25,205,90,125,16,251,6,8,197,6,255,16,254,219,255,23
50 DATA 210,83,125,218,76,125,6,40,205,90,125,16,251,193,205,30
60 DATA 2,16,229,123,205,51,0,211,253,195,12,125,203,3,203,131
70 DATA 195,54,125,203,3,203,195,195,54,125,197,6,62,16,254,193
80 DATA 201,62,83,50,255,63,205,73,0,205,51,0,211,253,254,1
90 DATA 202,0,125,254,64,200,95,6,8,203,123,204,136,125,196,161
100 DATA 125,203,3,16,244,195,102,125,197,6,28,62,5,211,255,197
110 DATA 6,50,16,254,62,6,211,255,6,50,16,254,193,16,236,193
120 DATA 201,197,6,46,205,90,125,16,251,193,201
130 A=32000:E=32170:FOR X=A TO E:READ B:POKEX,B:NEXT
135 RESTORE:C=0:FORX=A TO E:READ B:C=C+B:NEXT:C=C-20893
136 CLS: IFC<>0THENPRINT"FEHLER IN DATAZEILEN!":STOP
140 CLS:B=45.45:INPUT"BAUDRATE:";B:K=45.45/B
150 A=INT(255*K+0.5):V=INT(62*K+0.5):S=INT(40+(K*100-100)/(K*62)+0.5)
160 L0=INT(28*K+0.5):L1=INT(46+(K*300-300)/(K*62)+0.5)
170 POKE32042,A:POKE32092,V:POKE32055,S
180 POKE32138,L0:POKE32163,L1
190 POKE16526,0:POKE16527,125:D=USR(0)
200 GOTO140
```

Tabelle 2. BASIC-Programm für GENIE als ASCII-Fernschreiber. In den Zeilen 20 bis 130 wird das Maschinenprogramm geladen. Die Prüfzeilen 135 und 136 können nach dem ersten Durchlauf wieder gelöscht werden.

```
7D00: 3E 10 D3 FE CD 15 02 3E 45 32 FF 3F 1E FF CD 1E
7D10: 02 3A 40 38 FE 01 CA 61 7D DB FF 17 D2 11 7D 06
7D20: 19 CD 5A 7D 10 FB 06 08 C5 06 FF 10 FE DB FF 17
7D30: D2 53 7D DA 4C 7D 06 28 CD 5A 7D 10 FB C1 CD 1E
7D40: 02 10 E5 7B CD 33 00 D3 FD C3 0C 7D CB 03 CB 83
7D50: C3 36 7D CB 03 CB C3 C3 36 7D C5 06 3E 10 FE C1
7D60: C9 3E 53 32 FF 3F CD 49 00 CD 33 00 D3 FD FE 01
7D70: CA 00 7D FE 40 C8 5F 06 08 CB 7B CC 88 7D C4 A1
7D80: 7D CB 03 10 F4 C3 66 7D C5 06 1C 3E 05 D3 FF C5
7D90: 06 32 10 FE 3E 06 D3 FF 06 32 10 FE C1 10 EC C1
7DA0: C9 C5 06 2E CD 5A 7D 10 FB C1 C9 00 00 00 00 00
```

Tabelle 3. HEX-DUMP zum Maschinenprogramm

```
10 CLS:B=45.45:INPUT"Welche Baudrate:";B:K=45.45/B
20 A=INT(255*K+.5):V=INT(62*K+.5):S=INT(40+(K*100-100)/(K*62)+.5)
30 L0=INT(28*K+.5):L1=INT(46+(K*300-300)/(K*62)+.5)
40 POKE32042,A:POKE32092,V:POKE32055,S
50 POKE32138,L0:POKE32163,L1
60 POKE16526,0:POKE16527,125:D=USR(0)
70  GOTO10
```

Tabelle 4. BASIC-Programm zum Einstellen der BAUD-Rate

ringert werden. Ein Blick auf das BASIC-Programm zeigt, daß hierzu einige zusätzliche Berechnungen erforderlich sind, um einen Korrekturfaktor zu bestimmen, der die Laufzeit der Programmelemente ausgleicht, die nicht von den Zeitschleifen erfaßt werden.

Bei der Inbetriebname wird P1 im Interface bei angeschlossenem Computer so justiert, daß ohne Signal gerade 0 Volt auf der Fernleitung liegen. Der Empfangspegel wird mit P2 eingestellt. Mit einem Piezosummer, der parallel zur Fernleitung liegt, können die Signale kontrolliert werden. Auch als 'Klingel' ist er nützlich. □

Hardware für C-64

EBG 9600

Schnelles Kassettenlaufwerk für Daten- und Programmaufzeichnung, speziell Meßdatenerfassung. Übertragungsrate ca. 1 K Byte/sec. Bausatz mit Anschlußkabel (ohne Netzteil und Gehäuse)

Preis 680,– DM incl. MWSt.

EBG 2064

Epson-Interface zum Anschluß aller Epson-Drucker an den C-64

Preis 339,– DM incl. MWSt.

EBG 0488

Paralleles IEC-Interface: ermöglicht Anschluß an C-64, an CBM 4040, CBM 8050, Drucker, etc.

Preis 298,– DM incl. MWSt.

Rufen Sie uns an: (0 61 51) 31 38 90

EBG

Elektronik Bauelemente GmbH
Lothar Schanuel
Heidelberger Str. 73
6100 Darmstadt
Tel. (06151) 313890
Tx. 4197160

Was heißt 'CP/M-kompatibel'?

Holger Petersen

CP/M-Software läuft auf jedem CP/M-Rechner. Diese Aussage gilt für fast alle kommerziell angebotenen Programme. Probleme tauchen aber dann auf, wenn man Software selbst schreibt. Dann ist es notwendig, über die verschiedenen CP/M-Versionen und die jeweils verwendeten Floppy-Formate informiert zu sein. Dieser Artikel zeigt, wann ein Programm als kompatibel gelten darf und welche Bedingungen eingehalten werden müssen.

Die Frage, wann ein Programm oder ein Betriebssystem als 'CP/M-Kompatibel' gelten darf, ist von vornherein nicht einfach zu entscheiden. Das CP/M gibt es in verschiedenen, aufeinander aufbauenden Versionen, die jedoch auch nicht ganz aufwärtskompatibel sind.

CP/M-1.4 ist die Version, die bis circa 1980 überall in Betrieb war. CP/M-2.x ergab Fortschritte in bezug auf die Disketten- und Filegröße (nun bis 8 Megabyte statt 255 Kilobyte) und die Unterstützung des Video-Backspaces sowie der Einteilung der Diskette in Benutzerbereiche ('USER' 0—15). Zu CP/M-1.4, das immer noch auf einigen wenigen Geräten das einzige Betriebssystem ist, bleibt man nur kompatibel, wenn das gleiche Diskettenformat benutzt wird, man nur unter USER 0 arbeitet, keine File-Attribute anwendet, und ein File nicht länger als 256 Kbyte lang ist.

CP/M-3.x hat gegenüber CP/M-2.x ein neues File-Attribut (Archive) und DATUM-Einträge in der Directory. Sie bringen CP/M-2.x etwas aus dem Tritt; für auszutauschende Daten sollte man darauf verzichten.

Das CP/M-Betriebssystem stellt dem Programmierer eine Reihe von Systemfunktionen zur Verfügung (Tabelle 1). Wenn man mit diesen Funktionen nicht auskommt, kann man noch die BIOS-Sprungtabelle ausnutzen (nicht mehr für CP/M-3.x). Diese Technik hat sich eingebürgert, weil das 'alte' CP/M-1.4 keine Systemfunktion 'Input ohne Echo' besaß und weil die Umgehung des BDOS etwas schneller abläuft. Wenn der Character-Output auch über die BIOS-Sprungleiste abgewickelt wird, kann auch das Zeichen '07' (TAB) als solches ausgegeben werden, und die Steuerzeichen '13' (^S) und '10' (^P) können eingegeben werden, ohne daß das CP/M sie interpretiert. Man darf aber nur über die Adressen der Sprungvektoren bei 0000 und 0005 zugreifen. Ein Programm, das absolute Werte für die Sprungtabelle annimmt, ist ganz klar nicht transportabel.

Sehr viele Programme nehmen an, daß die BIOS-Sprungleiste an einer Page-Grenze beginnt. Wenn dem nicht so ist (KONTRON) gibt es in solchen Fällen Probleme.

Wenn nur die Werte für Console- und Printer-I/O benötigt werden, sollte man sich diese im Assembler-Programm ausrechnen (Bild 1). Werden auch noch die Disk-Funktionen benötigt, kann es sich empfehlen, die ganze Sprungtabelle zu kopieren (Bild 2). Sowohl bei BDOS-Aufrufen als auch bei den Sprungvektoren des BIOS sei ausdrücklich gesagt, daß alle Register verändert werden können. Achtung für Z 80-Programmierer: das gilt auch für die INDEX-Register und den zweiten Registersatz! Umgekehrt sollte derjenige, der sein BIOS selbst programmiert, diese Register dabei nicht verändern, um nicht in Schwierigkeiten mit einigen Anwenderprogrammen zu kommen (frühe Versionen von Interchange und Mychess). Bei angeblich 'kompatiblen' Betriebssystemen sollte man genau prüfen, ob der BDOS-Return-Code, wie bei CP/M, sowohl in den Registern A, BC und auch in HL zurückgegeben wird.

Z 80-Besitzer sollten beim Programmieren auch daran denken, daß eine Vielzahl von CP/M-Systemen 'nur' mit einer 8080- oder 8085-CPU betrieben wird. Wenn jemand die speziellen Z 80-Befehle benutzt, sollte er auch prüfen, ob das Programm auf den genannten CPUs abläuft. Es ist inzwischen fast selbstverständlich, daß man die Versionsnummer des verwendeten CP/M prüft und entweder eine Fehlermeldung ausgibt, oder sein Programmverhalten darauf einrichtet.

In der 'guten alten Zeit' des CP/M, vor der Version 1.4, gab es nur das von der Firma IBM abgeschaute Diskettenformat (77 Spuren, je 26 Sektoren zu je 128 Bytes). Davon werden zwei Tracks für das Betriebssystem selbst gebraucht, so daß eine Diskette dann 241 Kilobyte 'netto' enthält. Auch heute ist es noch so, daß alle CP/M-Computer, die ein 8"-Laufwerk haben, dieses Standardformat lesen können. Zwar gab es für CP/M-1.4 auch schon Implementationen für Mini-Laufwerke; mit dem Erscheinen von CP/M-2.x wurde es aber wesentlich einfacher, verschiedene Diskettenformate auf einem CP/M-Rechner zu 'fahren'. Da keine Firma schnell genug war, den 'Marktführer' zu spielen, ist es zu dem bedauerlichen Chaos der 5¼"-Diskettenformate gekommen. Es mag häufig auch mitgespielt haben, daß man den Kunden mit einem 'eigenen' Diskettenformat an sich binden wollte. Es gibt ja auch genug Variablen, mit denen man bei der Implementierung von CP/M-Disketten spielen kann:

Man kann 128, 256, 512 oder 1024 Bytes pro physikalischen Diskettensektor schreiben.

Durch die Wahl der Anzahl der Sektoren können zwischen 2048 (16*128) und 5632 (11*1024) Bytes pro Spur geschrieben werden.

Man kann 35 oder 40 Spuren (auf 96tpi-Laufwerken auch 70 oder 80) Spuren pro Disketttenseite nutzen. Man kommt so auf eine Diskettenkapazität, die zwischen 70 und 800 KByte pro 'Scheibe' liegt.

Für das Betriebssystem werden 1, 2, 3 oder 4 Spuren gebraucht. Sie sind nicht unbedingt in der gleichen Weise formatiert wie der Rest der Diskette.

Man sieht 32, 64, 128 oder 256 Directory-Entries vor.

Der Diskettenplatz wird in 1, 2 oder 4 KByte-Blöcke eingeteilt.

```
init:
        ld    hl,(1)          ; Lade Adresse des Warm-Start-Vektors
        ld    de,3            ; Länge eines Sprung-Befehls
        add   hl,de           ; nun zeigt HL auf Console-Status
        ld    (CSTAT+1),hl    ; eingesetzt
        add   hl,de           ; zeige auf Console-Input
        ld    (CINP+1),hl     ; einsetzen
        add   hl,de           ; zeige auf Console-Output
        ld    COUT+1),hl      ; einsetzen
        ret
;
CSTAT   jp    $               ; nach Ablauf der Routine 'INIT'
CINP    jp    $               ; stehen hier Sprünge, die direkt
COUT    jp    $               ; zu den BIOS-Routinen führen
;
```

Bild 1. Berechnung der Console-BIOS-Adressen

```
init:
        ld    hl,(1)          ; Lade Adresse des Warm-Start-Vektor's
        ld    de,table        ; Startadresse der programm-internen Tab.
        ld    bc,16*3         ; 16 Sprünge (CP/M-2.x)
                              ; sind zu kopieren
;
      ld      a,03h
      and     0fh             ; setzt bei Z80 das Half-Cary Flag
      daa                     ; macht das unterschiedliche Ver-
      cp      09              ; halten deutlich
      jp      z,Z80           ; sonst 8080
;
move80:
      ld      d,(hl)          ; hole Source-Byte
      ld      (de),a          ; hin zur Destination
      inc     hl
      inc     de
      dec     bc              ; 16 Bit INC/DEC setzen kein FLAG!
      ld      a,b
      or      c               ; nur NULL, wenn beide Register Ø
      jp      nz,move80       ; nochmal ...
;
      ret
;
; Z80-speziell LDIR
;
Z80:
        ldir                  ; kopiert den Bereich
        ret
;
TABLE:
;
WSTART  jp    $               ; hier stehen
CSTAT   jp    $               ; nach Ablauf
CINP    jp    $               ; der Routine
COUT    jp    $               ; 'INIT' die
LSTOUT  jp    $               ; BIOS-Routinen
READER  jp    $               ; direkt im
PUNCH   jp    $               ; Assembler-
HOME    jp    $               ; Listing als
SELDSK  jp    $               ; Label zur
SETTRK  jp    $               ; Verfügung
SETSEK  jp    $               ; Man beachte
SETDMA  jp    $               ; aber die
READ    jp    $               ; Parameter-
WRITE   jp    $               ; Konventionen
LSTST   jp    $               ; des CP/M
SEKTRN  jp    $               ;
;
```

Bild 2. Kopieren der Sprungleiste

Zusätzlich kann man den Sektor-Skew modifizieren oder die Sektoren invertiert auf die Diskette schreiben.

Alle Möglichkeiten vervielfachen sich, wenn man doppelseitige Laufwerke benutzt. Die Sektoren können auf der zweiten Seite weiternumeriert werden, oder von vorn beginnen, und so weiter.

Wer exotisch sein will, benutzt 77-Track-Laufwerke oder die neuen 5¼" Laufwerke, die in der Kapazität und Übertragungsrate kompatibel zu 8"-Laufwerken sind, und kommt dann auf 1,2 Megabyte. Von hard-sektorierten und anderen, nicht das IBM-soft-sektorierte Verfahren benutzenden Aufzeichnungsformaten (Apple, Commodore, Osi bzw. Sirius), soll gar nicht erst gesprochen werden. Diese Disketten sind wohl nur auf dem jeweiligen Originalrechner wieder zu lesen. Besitzer solcher Computertypen sind für den Datenaustausch auf die Benutzung von seriellen Schnittstellen angewiesen.

Bisher hat kaum eine Firma einen neuen CP/M-Rechner mit 5¼"-Laufwerken vorgestellt, der als Boot-Format ein schon existierendes Format benutzte (und wenn, war es wohl Zufall?). In der Tabelle 3 sind einige Vertreter von 5¼"-CP/M-Formaten vorgestellt. Erfreulich ist aber, daß immer mehr Firmen dazu übergehen, das Lesen und oft auch Schreiben auf einigen Fremdformaten zu ermöglichen. Pionier war Adam Osborne mit vier verschiedenen Formaten auf dem Osborn-I, die automatisch beim 'Einloggen' erkannt werden. Bei einigen anderen Rechnern muß man dem zweiten Laufwerk jeweils ein neues Diskettenformat zuordnen. Gerade für den OSBORN-Rechner gibt es aber ein Programm, um noch mehr Formate lesen zu können, mit dem beziehungsreichen Titel 'OZMOSIS'. Es ist aber auch schon im OSBORN-BIOS ein Platz vorgesehen, um ein weiteres Diskettenformat automatisch zu lesen. Die Adressen sind in der Tabelle 4 aufgeführt. Diese soll nicht als Kochrezept verstanden werden, sondern als Hilfe für diejenigen, die sich in BIOS-Modifikationen auskennen. →

Nummer	CP/M-1.4	C/PM-2.x	CP/M-3.x
0	Warmstart	dito	dito
1	Console Input	dito	dito
2	Console Output	dito	dito
3	Reader Input	dito	Aux. Input
4	Punch Output	dito	Aux. Output
5	List Output	dito	dito
6	Return NULL	Direct IO	dito
7	Interr. I/O-Byte	dito	Aux. Inp. Status
8	Set I/O-Byte	dito	Aux. Out. Status
9	Print String	dito	dito
10	Read Console Buffer	dito	dito
11	Console Status	dito	dito
12	Return NULL	get Version #	dito
13	Reset Disk System	dito	dito
14	Select Disk	dito	dito
15	Open File	dito	dito
16	Close File	dito	dito
17	Search First	dito	dito
18	Search Next	dito	dito
19	Delete File	dito	dito
20	Read Record	dito	dito
21	Write Record	dito	dito
22	Make File	dito	dito
23	Rename File	dito	dito
24	Return Login Vektor	dito	dito
25	Return Current Disk	dito	dito
26	Set DMA	dito	dito
27	Get Allocation Adr.	dito	dito
28	*Write Protect Disk	dito	dito
29	*Get R/O-Vektor	dito	dito
30	*Set Directory DMA	Set File Attribute	dito
31	%	Get DPB-Adresse	dito
32	%	Set/Get User-Number	dito
33	%	Read Random	dito
34	%	Write Random	dito
35	%	Compute File Size	dito
36	%	Set Random Record	dito
37	%	+ Reset Drive	dito
38	%	$	$
39	%	$	$
40	%	+ write random with Zero Fill	

* 28 — 30 unter CP/M-1.4 nicht dokumentiert
+ 37 & 40 unter CP/M-2.x nicht dokumentiert
$ 38 & 39 nur MP/M, unter CP/M 'Return NULL'
41—50, 59—60, 98—112 & 152 nur CP/M-3.x

Tabelle 1. System-Funktionen

Nummer	CP/M-1.4	C/PM-2.x	CP/M-3.x
xx00	Coldstart	dito	dito
xx03	Warmstart	dito	dito
xx06	Console Status	dito	dito
xx09	Console Input	dito	dito
xx0C	Console Output	dito	dito
xx0F	List Output	dito	dito
xx12	Reader Input	dito	Aux. Input
xx15	Punch Output	dito	Aux. Output
xx18	Home Disk Head	dito	don't use!
xx1B	select Disk	dito	don't use!
xx1E	select Track	dito	don't use!
xx21	select Sektor	dito	don't use!
xx24	Read 128 Bytes	dito	read phys. Sektor!
xx27	Write 128 Bytes	dito	write phys. Sektor!
xx30	nicht vorhanden	List Status	dito
xx33	nicht vorhanden	Sektor Translate	don't use!

Weitere Einträge nur unter CP/M-3.x definiert.

Tabelle 2. BIOS-Sprungleiste

Name	Bytes p. Sektor	Sektor p. Track	Tracks p. Disk	System-tracks	Kapazität	Sektor-Skew	Density	Block größe	Directory-Entries
TRS-80 R2	128	18	35	2	74	5	SD	1	64
Cromenco	128	18	35	2	74	4	SD	1	64
Xerox 820 SD	128	18	40	3	83	5	SD	1	32
Osborn SD	256	10	40	3	95	4	SD	2	64
Alphtronik PS	256	16	40	2	151	1	DD	1	64
IBM (CP/M-86)	512	8	40	1	156	1	DD	1	64
Superbrain	512	10	40	2	190	4	DD-inv.	2	64
Osborn DD	1024	5	40	3	188	1	DD	1	64

Tabelle 3. Einige typische Diskettenformate

```
            org     0E276h
Sektor-Translate Tabelle
entspricht Skew=1

            db      1,2,3,4,5,6,7,8,9
            db      10,11,12,13,14,15,16,17,18,19
            db      20,21,22,23,24,25,26,27,28,29
            db      30,31,32


            org     0E2EEh

            db      4           ; Drive-Code: 0000BB0D
                                ; D = 1 für SD, 0 für DD
                                ; BB = 00 für 128, 01 für 256,
                                ; 10 für 512, 11 für 1024 Bytes/Sektor
nun DPB:

            dw      32          ; log. Sektoren/Spur
            db      3           ; Sektor-Shift
            db      7           ; Sektormask
            db      0           ; Extent-Mask
            dw      150         ; Kapazität (—1)
            dw      0C00h       ; alloc
            dw      10h         ; Check-Size
            dw      2           ; res. Tracks
```

Tabelle 4. Lage der freien Disk-Parameter im OSBORN—Bios mit Beispiel für Alphatronik P2

⊙⊕ DAS UMFASSENDE PROGRAMM ⊕⊙

INTERFACES UND PUFFERSPEICHER FÜR DRUCKER

WIESEMANN 0202-510444

COMPUTER-SCHNITTSTELLE

DRUCKER-SCHNITTSTELLE	CBM HP-IB IEEE488	CBM64 VC20	Centronics	RS232C V24
CBM HP-IB IEEE488	BUFFER 8-120K ☞33□□□	Lieferant: Data Becker Düsseldorf	Interface OK (Jan.84) ☞2300	Interface OK (Nov.83) ☞8300
CBM64 VC20	Interface OK (Dez.83) ☞3900	BUFFER 8-120K (Jan.84) ☞99□□□	Interface OK ☞2900	Interface OK (Jan.84) ☞8900
Centronics	Interfaces 0-120K ☞32□□□	Interface OK ☞9200	BUFFER 8-120K ☞22□□□	Interfaces 0K-32K ☞82□□
RS232C V24	Interfaces 0-32K ☞38□□	Lieferant: Data Becker Düsseldorf	Interfaces 0-32K ☞28□□	BUFFER 8-120K ☞88□□□

ERLÄUTERUNG: Z.B.: 32□□□ = ARTIKELNUMMER
□□□ = BUFFERKAPAZITÄT

REINHARD WIESEMANN MIKROCOMPUTERTECHNIK
WINCHENBACHSTR. 3A POSTFACH 201605
5600 WUPPERTAL 2 TEL.: 0202/510444

CEPAC-65 Version A

Komplett-Bausatz o. EPROM
o. Steckerleiste ..... **DM 69,—** (fertig DM 85,—)

c't-86 16-bit-Selbstbaucomputer

Platine 1: CPU-Karte mit 8086, 8 KB Monitorprogramm
Komplett-Bausatz ... **DM 349,—** (fertig DM 449,—)

Platine 2: I/O-Karte mit V-24-, Centronics-, Kassettenrecorder-Interface und Timer
Komplett-Bausatz ... **DM 249,—** (fertig DM 349,—)

Platine 3: Floppy-Controller-Karte bis zu 4 Laufwerke 5¼ oder 8 Zoll (auch gemischt), IC WD 2797
Komplett-Bausatz ... **DM 498,—** (fertig DM 598,—)

Platine 4: 256 KB-RAM-Karte (128 K bestückt)
Komplett-Bausatz ... **DM 598,—** (fertig DM 698,—)

Busplatine (96polig)
10 Steckplätze ..... **DM 49,—**
fertig bestückt ..... **DM 169,—**

c't-Terminal

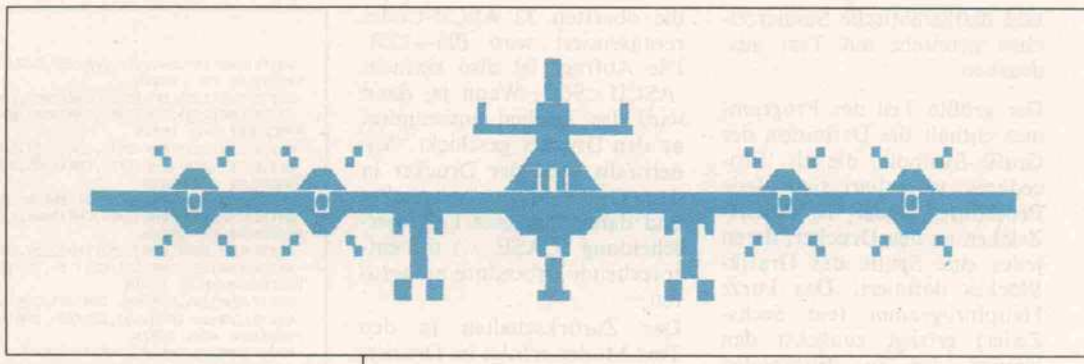
Version A Bausatz ..... **DM 449,—** (fertig DM 549,—)
Version B Bausatz ..... **DM 498,—** (fertig DM 639,—)

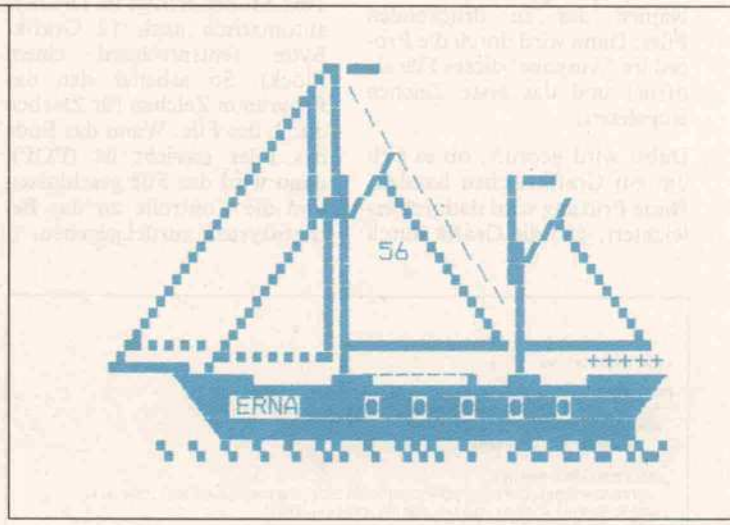
Versand per Nachnahme und Versandkosten.
Liste C3 anfordern.

Frölje Elektronik

Gaststr. 10 ★ 2900 Oldenburg ★ Tel. (04 41) 1 58 53

Eberhard Hewicker





Der OSBORNE 1 bietet zwar einen Blockgrafik-Satz von 32 Zeichen, aber den kann man nur mit Schwierigkeiten ausnutzen. Es wird entweder ein Grafik-Editor benötigt (ist im Handel erhältlich), oder man muß umständliche Assembler-Programme schreiben, um vor jeder Cursor-Bewegung den Grafik-Mode zu verlassen und anschließend wieder einzuschalten. Und das Assembler-Programmieren ist nun mal nicht jedermanns Sache.

WordStar als Grafik-Editor

Blockgrafik mit OSBORNE 1 und Epson MX 80

Die Anschaffung eines Grafik-Editors läßt sich elegant umgehen, indem man den mitgelieferten Editor, nämlich WordStar, benutzt. Die Idee ist einfach, die Wirkung verblüffend. Nach dem Laden von WordStar startet man mit dem Kommando 'R' (Run a Program) ein kleines Assembler-Programm, das die Umschaltung in den Grafik-Modus bewirkt. Dann geht man wieder zurück ins WordStar (durch Drücken einer beliebigen Taste). Die System-Meldungen sind jetzt teilweise durch Grafik-Zeichen entstellt, dadurch braucht man sich aber nicht beeindrucken zu lassen.

Mit dem Kommando 'N' wird ein non-document-file eröffnet. Auf dem Bildschirm werden jetzt alle Kleinbuchstaben als Grafik-Symbole dargestellt. Die vollen Editier-Möglichkeiten des WordStar stehen uneingeschränkt zur Verfügung. Man kann also z. B. Zeilen löschen oder einfügen, Blöcke umstellen, das File speichern und wieder laden usw. Für die Darstellung von Schrift ist man jetzt allerdings auf die Großbuchstaben angewiesen, denn die Kleinbuchstaben erzeugen ja die Grafik.

Nach Herzenslust kann man nun den Bildschirm vollmalen. Die Tab-Funktion sollte man allerdings vermeiden, denn das bringt den Drucker durcheinander, der schließlich das 'Gemälde' zu Papier bringen soll. In diesem Punkt ist das Druckprogramm noch verbesserungsfähig. Eine Überraschung wird man erleben, wenn man einen Block markieren will: statt halber Helligkeit erscheinen plötzlich wieder Kleinbuchstaben anstelle der Grafik. Durch ^KH kann aber die Grafik wiederhergestellt werden.

Am Ende der Sitzung wird das Grafik-File ganz normal abgespeichert (^KD), wobei die Floppy die Zeichen natürlich nach ASCII interpretiert und speichert. Das File kann erneut geladen und editiert werden wie jedes andere WordStar-File auch. Der Grafik-Modus wird durch 'Ausstieg zum System' ('X') wieder verlassen. Ein Wiedereinschalten ist auf die oben beschriebene Art jederzeit möglich.

Nun bleibt noch das Problem, die Grafik auch mit einem MX 80 zu Papier zu bringen, der ja von Hause aus keine Block-Grafik kann.

Die Lösung ist ein Pascal-Programm, welches das Grafik-File von einer Diskette einliest, für jedes Zeichen den entsprechenden Grafik-Code erzeugt und diesen dann an den Drucker ausgibt. Zu Beginn des Hauptprogramms wird der Zeilenvorschub so eingestellt, daß sich die Grafik-Blöcke nahtlos aneinanderfügen. Dies ist bei der Mischung von Grafik und Text zu beachten: Auf dem Bildschirm muß zwischen zwei Textzeilen immer eine Leerzeile eingefügt werden, sonst verfließen die Schriftzeichen beim Ausdrucken ineinander.

Da ein Drucker ohnehin nicht zu den schnellen Peripherie-Geräten gehört, wird der Druckvorgang durch das Pascal-Programm in keiner Weise verzögert. Natürlich könnte solch ein Druckprogramm auch in jeder anderen Programmiersprache geschrieben werden. Pascal erschien mir jedoch schnell, kompakt und gleichzeitig leicht verständlich. Hier wird der Compiler Hisoft HD verwendet.

Mit einem ähnlichen Programm lassen sich übrigens auch griechische Buchstaben

und mathematische Sonderzeichen gemischt mit Text ausdrucken.

Der größte Teil des Programmes enthält die Definition der Grafik-Symbole, die als 'Procedures' vereinbart sind. Jede Procedure sendet 12 ASCII-Zeichen an den Drucker, deren jedes eine Spalte des Grafik-Blockes definiert. Das kurze Hauptprogramm (ein Sechs-Zeiler) erfragt zunächst den Namen des zu druckenden Files. Dann wird durch die Procedure 'Ausgabe' dieses File eröffnet und das erste Zeichen eingelesen.

Dabei wird geprüft, ob es sich um ein Grafikzeichen handelt. Diese Prüfung wird dadurch erleichtert, daß die Grafik durch die obersten 32 ASCII-Codes repräsentiert wird (96—127). Die Abfrage ist also einfach: 'ASCII<96?'. Wenn ja, dann wird das Zeichen unverändert an den Drucker geschickt. Andernfalls wird der Drucker in den Grafik-Modus geschaltet und dann über eine Fallunterscheidung (CASE...) die entsprechende Procedure aufgerufen.

Das Zurückschalten in den Text-Modus erfolgt im Drucker automatisch nach 12 Grafik-Bytes (entsprechend einem Block). So arbeitet sich das Programm Zeichen für Zeichen durch das File. Wenn das Ende des Files erreicht ist (EOF), dann wird das File geschlossen und die Kontrolle an das Betriebssystem zurückgegeben. □

```
type GRAPHIK.PAS

PROGRAM Graphik;{Fuer Block-Graphik mit EPSON MX80}
                {Eberhard Hewicker, Koblenz, 1983 }
 VAR Name :ARRAY[1..14] OF CHAR;
     Data :FILE OF CHAR;

 PROCEDURE KA; BEGIN
  WRITE(CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(0));
  WRITE(CHR(0),CHR(0),CHR(0),CHR(0),CHR(0));END;
 PROCEDURE KB; BEGIN
  WRITE(CHR(15),CHR(15),CHR(15),CHR(15),CHR(15),CHR(15));
  WRITE(CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(240));END;
 PROCEDURE KG;BEGIN
  WRITE(CHR(0),CHR(0),CHR(0),CHR(0),CHR(0),CHR(0),CHR(15));
  WRITE(CHR(15),CHR(15),CHR(15),CHR(15),CHR(15));END;
 PROCEDURE KD;BEGIN
  WRITE(CHR(0),CHR(0),CHR(0),CHR(0),CHR(0),CHR(0),CHR(255));
  WRITE(CHR(255),CHR(255),CHR(255),CHR(255),CHR(255));END;
 PROCEDURE KI;BEGIN
  WRITE(CHR(0),CHR(0),CHR(0),CHR(63),CHR(63),CHR(63),CHR(63),CHR(63));
  WRITE(CHR(63),CHR(60),CHR(60),CHR(60));END;
 PROCEDURE KE;BEGIN
  WRITE(CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(255));
  WRITE(CHR(255),CHR(255),CHR(255),CHR(255),CHR(255));END;
 PROCEDURE KF;    BEGIN
  WRITE(CHR(15),CHR(15),CHR(15),CHR(15),CHR(15),CHR(0));
  WRITE(CHR(0),CHR(0),CHR(0),CHR(0),CHR(0));END;
 PROCEDURE KJ;    BEGIN
  WRITE(CHR(1),CHR(1),CHR(3),CHR(3),CHR(7),CHR(15),CHR(15));
  WRITE(CHR(31),CHR(63),CHR(63),CHR(127),CHR(255));END;
 PROCEDURE KH;    BEGIN
  WRITE(CHR(255),CHR(127),CHR(63),CHR(63),CHR(31),CHR(15),CHR(15));
  WRITE(CHR(7),CHR(3),CHR(3),CHR(1),CHR(1));END;
 PROCEDURE KL; BEGIN
  WRITE(CHR(60),CHR(60),CHR(60),CHR(255),CHR(255),CHR(255),CHR(255));
  WRITE(CHR(255),CHR(255),CHR(60),CHR(60),CHR(60));END;
 PROCEDURE KO; BEGIN
  WRITE(CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(240));
  WRITE(CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(240));END;
 PROCEDURE KM; BEGIN
  WRITE(CHR(0),CHR(0),CHR(60),CHR(126),CHR(126),CHR(126),CHR(126),CHR(126));
  WRITE(CHR(126),CHR(60),CHR(0),CHR(0));END;
 PROCEDURE KN; BEGIN
  WRITE(CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(15));
  WRITE(CHR(15),CHR(15),CHR(15),CHR(15),CHR(15));END;
 PROCEDURE KO; BEGIN
  WRITE(CHR(60),CHR(60),CHR(60),CHR(63),CHR(63),CHR(63),CHR(63),CHR(63));
  WRITE(CHR(63),CHR(60),CHR(60),CHR(60));END;
 PROCEDURE KU; BEGIN
  WRITE(CHR(128),CHR(128),CHR(192),CHR(192),CHR(224),CHR(240),CHR(240));
  WRITE(CHR(248),CHR(252),CHR(252),CHR(254),CHR(255)); END;
 PROCEDURE KP;    BEGIN
  WRITE(CHR(60),CHR(60),CHR(60),CHR(63),CHR(63),CHR(63),CHR(63));
  WRITE(CHR(63),CHR(63),CHR(0),CHR(0),CHR(0));END;
 PROCEDURE KW; BEGIN
  WRITE(CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(240));
  WRITE(CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(240));END;
 PROCEDURE KR; BEGIN
  WRITE(CHR(240),CHR(240),CHR(240),CHR(240),CHR(240),CHR(0),CHR(0));
  WRITE(CHR(0),CHR(0),CHR(0),CHR(0));END;
 PROCEDURE KS;  BEGIN
  WRITE(CHR(1),CHR(3),CHR(7),CHR(15),CHR(31),CHR(62),CHR(124),CHR(248));
  WRITE(CHR(240),CHR(224),CHR(192),CHR(128));END;
 PROCEDURE KX;  BEGIN
  WRITE(CHR(15),CHR(15),CHR(15),CHR(15),CHR(15),CHR(15),CHR(15));
  WRITE(CHR(15),CHR(15),CHR(15),CHR(15),CHR(15));END;
 PROCEDURE KV;  BEGIN
  WRITE(CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(255));
  WRITE(CHR(255),CHR(255),CHR(255),CHR(255),CHR(255));END;
 PROCEDURE KY;  BEGIN
  WRITE(CHR(255),CHR(254),CHR(252),CHR(252),CHR(248),CHR(240),CHR(240));
  WRITE(CHR(224),CHR(192),CHR(192),CHR(128),CHR(128));END;
 PROCEDURE KT; BEGIN
  WRITE(CHR(0),CHR(0),CHR(0),CHR(0),CHR(0),CHR(0),CHR(240));
  WRITE(CHR(240),CHR(240),CHR(240),CHR(240),CHR(240));END;
 PROCEDURE KZ; BEGIN
  WRITE(CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(255),CHR(15));
  WRITE(CHR(15),CHR(15),CHR(15),CHR(15),CHR(15));END;
 PROCEDURE KKLA; BEGIN
  WRITE(CHR(60),CHR(60),CHR(60),CHR(60),CHR(60),CHR(60),CHR(60));
  WRITE(CHR(60),CHR(60),CHR(60),CHR(60),CHR(60));END;
 PROCEDURE KDA; BEGIN
  WRITE(CHR(0),CHR(0),CHR(0),CHR(252),CHR(252),CHR(252),CHR(252));
  WRITE(CHR(252),CHR(252),CHR(60),CHR(60),CHR(60));END;
 PROCEDURE KTILDE; BEGIN
  WRITE(CHR(60),CHR(60),CHR(60),CHR(252),CHR(252),CHR(252),CHR(252));
  WRITE(CHR(252),CHR(252),CHR(60),CHR(60),CHR(60));END;
 PROCEDURE GRAFIK; BEGIN
  WRITE(CHR(27),'L',CHR(12),CHR(128));END;{meldet dem MX80 12 Grafikworte}
 PROCEDURE KK; BEGIN
  WRITE(CHR(0),CHR(0),CHR(0),CHR(255),CHR(255),CHR(255),CHR(255));
  WRITE(CHR(255),CHR(255),CHR(0),CHR(0),CHR(0));END;
 PROCEDURE KC;    BEGIN
  WRITE(CHR(15),CHR(15),CHR(15),CHR(15),CHR(15),CHR(15),CHR(255));
  WRITE(CHR(255),CHR(255),CHR(255),CHR(255),CHR(255));END;
 PROCEDURE KKLZ; BEGIN
  WRITE(CHR(128),CHR(192),CHR(224),CHR(240),CHR(248),CHR(124),CHR(62));
  WRITE(CHR(31),CHR(15),CHR(7),CHR(3),CHR(1));END;



 PROCEDURE Ausgabe;
  BEGIN
   RESET(Data,Name);PRON;    {File eroeffnen, Drucker ein}
   WRITE(CHR(27),CHR(51),CHR(25)); {Zeilenabstand=25/216 inch}
   WHILE NOT EOF(Data) DO                  {End of File?}
     BEGIN                                 {wenn ja, dann END!}
      GET(Data);                           {Lies ein Zeichen}
      IF EOLN(Data) THEN WRITELN  {wenn Zeilenende, dann 1 Zeilenvorschub}
        ELSE BEGIN;
          IF (Data^<'a') THEN WRITE(Data^){Normalzeichen, zum Drucker}
           ELSE BEGIN                      {Sonderzeichen, Procedure suchen}
           CASE Data^ OF
            'a':BEGIN GRAFIK;KA END;     'b':BEGIN GRAFIK;KB END;
            'g':BEGIN GRAFIK;KG END;     'd':BEGIN GRAFIK;KD END;
            'i':BEGIN GRAFIK;KI END;     'e':BEGIN GRAFIK;KE END;
            'f':BEGIN GRAFIK;KF END;     'j':BEGIN GRAFIK;KJ END;
            'h':BEGIN GRAFIK;KH END;     'l':BEGIN GRAFIK;KL END;
            'u':BEGIN GRAFIK;KU END;     'm':BEGIN GRAFIK;KM END;
            'n':BEGIN GRAFIK;KN END;     'o':BEGIN GRAFIK;KO END;
            'w':BEGIN GRAFIK;KW END;     'p':BEGIN GRAFIK;KP END;
            'x':BEGIN GRAFIK;KX END;     'k':BEGIN GRAFIK;KK END;
            'r':BEGIN GRAFIK;KR END;     's':BEGIN GRAFIK;KS END;
            'y':BEGIN GRAFIK;KY END;     'v':BEGIN GRAFIK;KV END;
            'z':BEGIN GRAFIK;KZ END;     'c':BEGIN GRAFIK;KC END;
            't':BEGIN GRAFIK;KT END;     'q':BEGIN GRAFIK;KQ END;
            '{':BEGIN GRAFIK;KKLA END;   '|':BEGIN GRAFIK;KDA END;
            '~':BEGIN GRAFIK;KTILDE END; '}':BEGIN GRAFIK;KKLZ END
           END
          END
        END
     END;



 BEGIN    {Beginn des Hauptprogrammes}
   WRITELN;WRITE('Eingabe des Filenamens:');
   READLN;
   READ(Name); {Lies Namen des zu druckenden Files. Name muss 14 Zeichen}
  Ausgabe;     {lang sein, gegebenenfalls durch SPACES auffuellen!      }
 END.          {z.B.: A:TEST____.PAS  oder: __TEST____.PAS              }



type B:GRAFIK.ASM
        escape  equ 1BH ;escape character
        graph   equ 'g' ;graphics
        pbuff   equ  9  ;BDOS print buffer
        bdos    equ  05H;BDOS entry

        org     0100H
        LXI     D,STARGR
        CALL    PRINT
        RET
 PRINT: MVI     C,PBUFF
        JMP     BDOS
STARGR: DB escape,graph,'$'

        END
```

Farbig plotten und drucken mit der X100 Serie von Adcomp.

Der X100S Superintelligenz und Präzision

Damit Sie nicht mit Routinearbeit belastet werden. Und – damit Ihr Rechner frei wird für andere Aufgaben.

20 kbyte Spool Puffer
Alle Schriftrichtungen
Ellipsen und Kreise
Schraffierte Flächen
Logarithmische Achsen
Kubische Interpolation für weiche Kurven
Barcode und Hardcopy
Schnittstellen nach Wahl:
RS 232 C (V24) mit Prozedur
XON XOFF auch für DEC.
IEEE 488 für HP und CBM
Centronics parallel

Modernste Technik:
2 Mikroprozessoren für Zeitoptimierung
Plotten mit 12 cm/sek.
Drucken mit 8 Z/sek.
Format DIN A4 Endlos und Einzelblatt
4 feine Farbminen im Revolver
Präziser und schneller Farbwechsel
Plottauflösung max. 0,05 mm

Plotter-Printer X100S

Die Exclusiv-Version X100S kostet DM 4.104,– incl. MWSt.

Andere Versionen der X100 Serie sind ab DM 2.394,– incl. MWSt. zu haben.

adcomp

adcomp Datensysteme GmbH
Olgastraße 15, 8000 München 19
Tel. 089/129 80 45, Telex 5 216 271

Hannover-Messe
Halle 4 · Stand 1302/I

Soll ein Computer Vorgänge aus der 'analogen Welt' erfassen, überwachen oder regeln, so müssen diese in die Digitalform übersetzt werden. In diesem Praxistip stellen wir einen einfachen und sehr preisgünstigen Analog/Digital-Wandler vor, der sich zur genauen Umsetzung von Spannungswerten eignet. Ein solcher Wandler ist vor allem für die Erfassung von Meßdaten über die verschiedensten Meßwertaufnehmer nützlich. Es lassen sich beispielsweise Temperaturfühler, Drucksensoren oder fotoelektrische Aufnehmer anschließen. Assemblerprogramme für Z 80 und 6502 ermöglichen die Übergabe der Meßdaten an den Computer.

Wolfgang Schrader

Einfacher A/D-Wandler

CA 3162 in Z 80- und 6502-Systemen

Als Wandler-Baustein wird der CA 3162 verwendet, ein IC, das man häufig in preisgünstigen Digital-Meßgeräten vorfindet. Der Baustein ist so konzipiert, daß man mit einem nachgeschalteten Dekoder/Anzeigetreiber und wenigen externen Bauelementen ein einfaches Voltmeter aufbauen kann. Der Wandler ist zwar nicht speziell für die Zusammenarbeit mit Mikrocomputersystemen gedacht, doch lassen seine Vorzüge (nicht zuletzt der geringe Preis von rund 5 Mark) auch diese Verwendung sinnvoll erscheinen.

Mit dem CA 3162 wird eine Auflösung von 1098 Punkten (+999 mV...—99 mV) erreicht, also besser als 2^{10} Bit. Der Wandler arbeitet im 'Dual-Slope'-Verfahren und führt wahlweise vier oder 96 Messungen pro Sekunde aus.

Zum Anschluß an den Mikrocomputer werden die Steuerleitungen für die drei Ziffern (Digits) der Anzeige zweckentfremdet. Sie zeigen an, zu welchem Zeitpunkt an den vier Datenleitungen der Wert für die Hunderter-, Zehner- und Einerstelle im BCD-Code ausgegeben wird. Die entsprechende Steuerleitung führt dann log. 0. Zum Anschluß an einen Computer werden also lediglich sieben Datenleitungen benötigt.

Der Wandler gibt den Meßwert in der Reihenfolge Hunderter, Einer, Zehner in ununterbrochener Folge aus. Die abgedruckten Assemblerprogramme für Z 80 (8080) und 6502 arbeiten beide nach demselben Prinzip: Der Computer wartet auf das Aktivieren einer Steuerleitung, um dann den Wert für die jeweilige Stelle in einen Zwischenspeicher zu übertragen. Aus diesem kann der Meßwert beispielsweise an ein BASIC-Programm übertragen werden.

Bei beiden Programmvorschlägen wurde der Eleganz wegen ein kleiner Trick verwendet: Der Computer überträgt den eingelesenen Wert schon in den Puffer, bevor er die 'zuständige' Steuerleitung abfragt. So geht man sicher, das Meßwert und Steuerleitung exakt zu demselben Zeitpunkt gelesen werden, ohne einen weiteren Zwischenspeicher einsetzen zu müssen.

Die Unterprogramme enthalten keine absoluten Sprünge und können deshalb an beliebiger Stelle im Speicher untergebracht werden. Lediglich die Lage des Zwischenspeichers sowie die Adresse des Eingangsports sind rechnerspezifisch und müssen den Gegebenheiten angepaßt werden. Als Port wird in unserem Schaltungsvorschlag (Bild 1) ein 8-Bit-Pufferbaustein (IC 1) verwendet. Wenn Ihr Computer über einen User-Port verfügt, können Sie auf den Puffer und den Adreßdekoder IC 3 verzichten und den Wandler direkt anschließen.

Natürlich könnte man theoretisch auch ein entsprechendes BASIC-Programm zur Abfrage eines Eingangsports verwenden. Allerdings wird das bei schnell wechselnden Meßwerten gelegentlich zu Ablesefehlern führen. So kann bei einem langsamen Interpreter folgender Effekt auftreten: Die Anzeige wechselt von 600 auf 599; der Rechner liest die Hunderterstelle (6), doch noch bevor die Zehner- und Einerstellen gelesen werden, gibt der Wandler den nächsten Wert aus. Als Zehnerstelle wird nun die '9' eingelesen. So entstehen Meßfehler, die vor allem bei grafischer Ausgabe der Daten einen unschönen 'Ausrutscher' darstellen. Ein Maschinenprogramm eignet sich deshalb besser.

Die Routine schreibt bei jedem Durchlauf alle drei Werte in der Reihenfolge Hunderter — Einer — Zehner in einen Zwischenspeicher. Auf diese Weise ist eine problemlose Zusammenarbeit zum Beispiel mit einem BASIC-Programm gewährleistet, das die Werte zur weiteren Bearbeitung nur noch durch PEEK auslesen muß.

Die Schaltung kann ohne

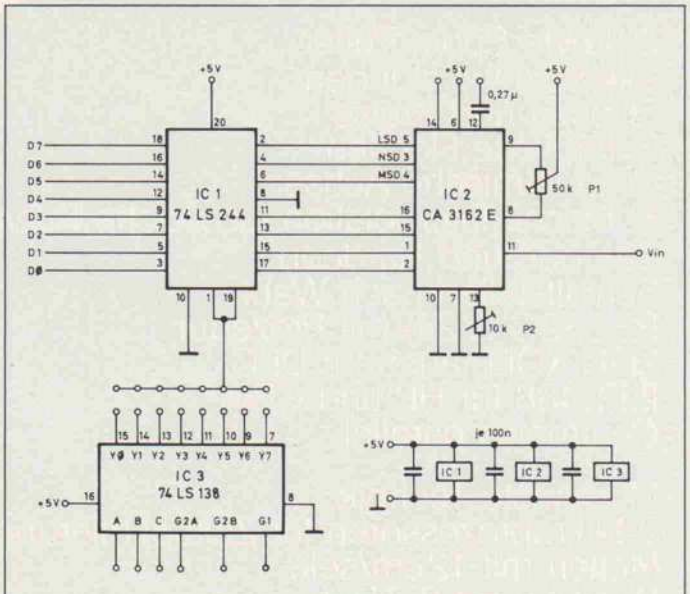


Bild 1. Pufferbaustein und Dekoder werden nur dann benötigt, wenn kein freier User-Port zum Anschluß des Wandler-ICs vorhanden ist.

Schwierigkeiten auf einer kleinen Lochrasterplatine aufgebaut werden. Der Abgleich des CA 3162 erfolgt mit den beiden Trimmpotis. Der Nullpunkt wird mit P1 eingestellt. Zum weiteren Abgleich legt man eine Spannung bekannter Höhe (möglichst 900 mV) an und stellt P2 so ein, daß dieser Meßwert ausgegeben wird. Der Regelbereich liegt bei ± 50 mV.

An Pin 6 des Wandlers kann die 'Conversion Rate' eingestellt werden. Legt man Pin 6 direkt an +5 V, führt der Wandler 96 Messungen in der Sekunde aus; legt man den Anschluß auf Masse, sind es vier Messungen. Wird Pin 6 über einen 12-K-Widerstand mit +5 V verbunden, so hält der Wandler den letzten Meßwert.

Die Eingangsimpedanz beträgt 100 MOhm. Über entsprechende Spannungsteiler können beliebige Meßbereiche eingestellt werden. Bei Überlauf in positiver Richtung (> +999 mV) gibt der Wandler eine '11' (0Bh) an jeder Stelle aus, bei Überschreitung des Meßbereichs in negativer Richtung (< —99 mV) eine '10' (0Ah). Ein negativer Meßwert wird außerdem durch Ausgabe einer '10' in der Hunderterstelle angezeigt.

Noch ein Wort zur Adreßdekodierung bei Verwendung des Pufferbausteins und des Adreßdekoders: Der Puffer muß bei einem Lesezugriff mit einem Selektionssignal (log. 0) aktiviert werden. Mit dem Dekoder 74LS138 läßt sich zwar keine vollständige, aber eine für viele Anwendungsfälle ausreichende Dekodierung erreichen. Tabelle 1 ist die Wahrheitstabelle des Dekoders, anhand derer Sie den Anschluß vornehmen sollten. Zu beachten ist, daß der Dekoder zwei Eingänge 'G2' besitzt, die miteinander 'ODER-verknüpft' sind. Beim Einsatz in einem Z 80-System sollte man hier die Signale $\overline{IORQ}$ und $\overline{RD}$ anlegen. □

```
                        ;*********************************************************
                        ;***    A/D-WANDLER SERVICE-ROUTINE                    ***
                        ;*********************************************************
                        ;
                        .Z80
                        ORG 100H
                        ;
0100'   F5              HUND:   PUSH AF                 ;REGISTER RETTEN
                        ;
                        ;A/D-WANDLER SIGNALISIERT VORHANDENSEIN DER JEWEILIGEN
                        ;BCD-STELLE DURCH LOW-PEGEL DER ZUGEHOERIGEN SIGNALLEITUNG
                        ;
                        ;
0101'   DB FF           HUND:   IN A,(0FFH)             ;HUNDERTERSTELLE EINLESEN
0103'   E6 2F                  AND 02FH                ;WENN SIGNALLEITUNG (BIT 5)
                                                        ;=0, DANN IST JETZT DAS
                                                        ;OBERE NIBBLE GELOESCHT
0105'   32 0124'                LD (HUN),A             ;HUNDERTERSTELLE SPEICHERN
0108'   E6 20                   AND 020H               ;WAR BIT 5 WIRKLICH 0?
010A'   20 F5                   JR NZ,HUND             ;WENN NICHT, NEUER VERSUCH
                        ;
010C'   DB FF           EINER:  IN A,(0FFH)            ;CA 3162 GIBT JETZT EINER-
                                                        ;STELLE AUS
010E'   E6 8F                   AND 08FH               ;OBERES NIBBLE LOESCHEN
0110'   32 0126'                LD (EIN),A             ;EINERSTELLE SPEICHERN
0113'   E6 80                   AND 080H               ;WAR BIT 7 WIRKLICH 0?
0115'   20 F5                   JR NZ,EINER            ;WENN NICHT, NEUER VERSUCH
                        ;
0117'   DB FF           ZEHNER: IN A,(0FFH)            ;ZEHNERSTELLE EINLESEN
0119'   E6 4F                   AND 04FH               ;OBERES NIBBLE LOESCHEN
011B'   32 0125'                LD (ZEHN),A            ;ZEHNERSTELLE SPEICHERN
011E'   E6 40                   AND 040H               ;WAR BIT 6 WIRKLICH 0?
0120'   20 F5                   JR NZ,ZEHNER           ;WENN NICHT↑ NEUER VERSUCH
                        ;
0122'   F1                      POP AF                 ;REGISTER ZURUECK
0123'   C9                      RET
                        ;
0124'   00              HUN:    DEFB    00H            ;SPEICHERZELLEN JETZT
0125'   00              ZEHN:   DEFB    00H            ;IN DURCHGAENGIGER
0126'   00              EIN:    DEFB    00H            ;REIHENFOLGE
                        ;
                        END
```

Tabelle 2. Abfrageprogramm für 8080 und Z 80

Eingänge					Ausgänge							
Enable		Select										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Positive Logik
* G2 = G2A + G2B

Tabelle 1. Wahrheitstabelle des Adreßdekoders 74LS138

```
                ;*****************************************************
                ;***   A/D-WANDLER SERVICEROUTINE FUER 6502
                ;*****************************************************
                ;
                MEM=?           ;DREI SPEICHERSTELLEN FUER EINGELESENE
                                ;WERTE (HUNDERTER, ZEHNER, EINER)
                PAD=?           ;EINGANGSPORT
                *=?             ;PROGRAMMANFANGSADRESSE
                ;
                ;
                LDY #00         ;Y-REGISTER INITIALISIEREN
                ;
HUND:   LDA PAD         ;PORTREGISTER LESEN
        AND #%00011111  ;WENN SIGNALLEITUNG (HUND)
                        ;=0, DANN IST BIT 5 GELOESCHT
        STA MEM,Y       ;HUNDERTER-WERT ABSPEICHERN
        AND #%00010000  ;WAR BIT 5 WIRKLICH 0?
        BNE HUND        ;WENN NICHT, NEUER VERSUCH
        ;
        LDY #02         ;INDEX FUER EINERSTELLE
        ;
EINER:  LDA PAD         ;CA 3162 GIBT JETZT
                        ;EINERSTELLE AUS
        AND #%10001111  ;OBERES NIBBLE LOESCHEN
        STA MEM,Y       ;EINER-WERT ABSPEICHERN
        AND #%10000000  ;WAR BIT 7 WIRKLICH 0?
        BNE EINER       ;WENN NICHT, NEUER VERSUCH
        ;
        DEY             ;INDEX FUER ZEHNERSTELLE
        ;
ZEHNER  LDA PAD         ;ZEHNERSTELLE EINLESEN
        AND #%01001111  ;OBERES NIBBLE LOESCHEN
        STA MEM,Y       ;ZEHNERSTELLE SPEICHERN
        AND #%01000000  ;WAR BIT 6 WIRKLICH 0?
        BNE ZEHNER      ;WENN NICHT, NEUER VERSUCH
        ;
        RTS
```

Tabelle 3. Abfrageprogramm für 6502

Super-Chips

Hans-Joachim Gelke

Der NS32032 ist augenblicklich der leistungsfähigste Baustein der NS16000-Mikroprozessorfamilie von National Semiconductor. Diese Familie wurde für einen hohen Datendurchsatz entwickelt, wie er zum Beispiel in Minicomputern, CAD-Arbeitsplätzen, Prozeßrechnern und für Telekommunikationsanwendungen benötigt wird. Man verfolgte strikt die Verwirklichung des Konzeptes, Eigenschaften von Großcomputern, wie Hochsprachenprogrammierung und virtuelles Speicherkonzept, für Mikroprozessoren zu übernehmen. Auf diese Weise entstand eine Mikroprozessoren-Familie einer völlig neuen Generation.

Die NS16000-Familie besteht aus mehreren unterschiedlich leistungsfähigen CPUs und einer Auswahl von Slaveprozessoren sowie Peripheriebausteinen (Bild 1).

Prozessoren

Bisher sind drei verschiedene CPUs erhältlich. Das Besondere daran ist, daß alle drei eine völlig identische interne Architektur haben, das heißt jeder Prozessor hat beispielsweise interne 32-Bit-Datenwege, 32-Bit-Architektur, die gleiche Anzahl von Registern und die gleiche Speicherorganisation.

Das bedeutet, ein Programm, das für einen 'kleinen' NS16000 entwickelt wurde, läuft auch auf einem 'Großen'. Bei der schnell fortschreitenden Technologie und angesichts der hohen Softwarekosten ist das ein großer Vorteil.

Die CPUs unterscheiden sich eigentlich nur nach außen hin, nämlich in der Breite des externen Datenbus'. Es existieren CPUs mit einem externen Datenbus von 8 Bit (NS08032), 16 Bit (NS16032) und 32 Bit (NS32032).

Neben den CPUs gibt es noch sogenannte Slaveprozessoren, die bei Bedarf ohne Softwareveränderung eingesetzt werden können, da sie für den Programmierer unbemerkt arbeiten.

Sie übernehmen Aufgaben wie zum Beispiel exakte und schnelle Fließkommaberechnungen und virtuelle Speicherverwaltung. Ein Begriff, der später etwas ausführlicher erklärt wird.

Mit der Floating Point Unit (FPU) NS16081 lassen sich schnelle Fließkommaberechnungen durchführen. Es kann eine Genauigkeit von 32 Bit oder 64 Bit gewählt werden.

Wird die CPU nicht mit dem Arithmetik-Slaveprozessor erweitert, können Fließkommaberechnungen auch softwaremäßig durchgeführt werden. Die Befehle zur Ausführung der Gleitkommarechnungen bleiben gleich, deshalb muß die Anwender-Software nicht geändert werden.

Dies ist möglich mit Hilfe der Modularen Software, die ich in einem späteren Abschnitt noch genauer beschreibe.

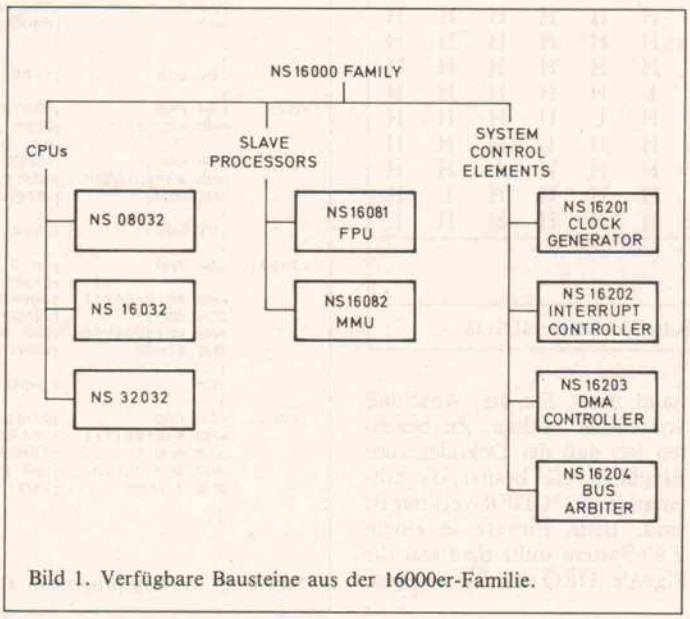


Bild 1. Verfügbare Bausteine aus der 16000er-Familie.

Peripheriebausteine

Die CPU NS32032 besitzt zwei Hardwareinterrupt-Eingänge, einen für maskierten und einen für nichtmaskierten Interrupt. Durch Anschluß einer oder mehrerer Interrupt-Steuerbausteine (Interrupt Control Unit), lassen sich diese Eingänge erweitern. Mit einer ICU sind bis zu 16 Interrupt-Eingänge verfügbar. Diese Eingänge sind mit bis zu 16 ICUs kaskadierbar. Somit erhält man bei Bedarf bis zu 256 Interrupt-Eingänge.

Das Taktsignal, bis zu 10 MHz, wird mit Hilfe eines externen Schaltkreises, genannt Timing-Control-Unit, kurz TCU, erzeugt. Mit ihm werden auch die Read- und Write-Zyklen gesteuert. Mit seiner Hilfe läßt sich der Bus-Zugriff bei Bedarf für eine vorprogrammierbare Zeit anhalten, zum Beispiel für langsame Speicher. In Entwicklung ist auch ein DMA-Controller für direkten Speicherzugriff.

Entworfen für Hochsprachen

Die Architektur beschreibt dem Anwenderkreis, der die Programme erstellt, wie der Computer aussieht. Sie beschreibt, welche und wie viele Register er hat, wie auf Speicherplätze zugegriffen werden kann, wie die Arithmetik organisiert ist, der Befehlssatz aufgebaut ist usw. Sie legt damit die Schnittstelle zwischen Software und Computer fest. Auf die interne Architektur wird etwas später genauer eingegangen.

Die Erfahrung von Programmherstellern hat gezeigt, daß sich Software für Mikroprozessoren in Hochsprachen (Pascal, Fortran, ADA) schneller erstellen und testen läßt als in Assemblersprache. Deshalb geht der Trend dahin, Programme zunächst mit Hilfe von Hochsprachen zu entwickeln und danach in die Maschinensprache des Prozessors zu übersetzen. Dieser Schritt erfolgt mit Hilfe von sogenannten Compilern.

Bei den gewöhnlichen Maschinensprachen und Prozessorarchitekturen entsteht dabei oft das Problem, daß sich ein Befehl nicht ohne weiteres übersetzen läßt. Man spricht dann, wie auch bei zwei Völkern mit unterschiedlichen Sprachen,

von einem Semantic Gap oder einer Übersetzungslücke. Um diese Lücken zu überbrücken, benötigt man meist lange und aufwendige Programmteile, die sich ungünstig auf die Rechengeschwindigkeit auswirken.

Aus diesen Gründen ist die Architektur der NS16000-Serie so ausgelegt, daß sie Hochsprachen wie zum Beispiel Pascal oder C entgegenkommt. Dadurch ergeben sich wesentlich kürzere Maschinencodes und damit schnellere Rechenzeiten. Die Compiler werden einfacher und kleiner.

Ein Programmbeispiel (Bild 2) zeigt die Verschiebung von 16 Datenbytes von einer Quelle (SOURCE) zu einem Ziel (DESTIN).

1. Im 8080-Assembler-Code
2. In der Programmiersprache Pascal
3. Im Assembler Code des NS32032

Datentypen

A + B = B
Operand 1 Operand 2 Ergebnis

Bei den Prozessoren der NS16000-Serie können für die Operanden unterschiedliche Bit-Längen gewählt werden. Man nennt dies Datentypen. Die beim NS32032 möglichen Datentypen zeigt Bild 3.

Bei Fließkommaberechnungen werden die Zahlen in Exponent und Mantisse aufgeteilt, zum Beispiel

$0{,}0473 = 4{,}73 \times 10^{-2}$

oder

$3970{,}54 = 3{,}97054 \times 10^{3}$

Adressierungsarten

Um den Inhalt einer Speicherzelle auslesen zu können, zum Beispiel um mit ihrem Inhalt und dem Inhalt einer anderen Speicherzelle eine Multiplikation durchzuführen, muß man zunächst einmal ihre Adresse bestimmen. Beim NS32032 gibt es hierfür neun Möglichkeiten. Die üblichen drei Methoden sind:

Direkte Adressierung:
Der Wert des Operanden steht direkt im Programm.

Absolute Adressierung:
Die Adresse der Speicherstelle, die als Operand dient, steht im Programm.

ⓐ
```
program MOVE;
var SOURCE, DESTIN: array[1...16] of char;
begin
  DESTIN := SOURCE
end.
```

ⓑ
```
        TITLE MOVE
        CSEG              ;CODESEGMENT:
LENGTH  EQU   16          ;DEFINE LENGTH.
MOVE:   MVI   HL,SOURCE   ;LOAD HL WITH ADDRESS OF 'SOURCE'.
        MVI   DE,DESTIN   ;LOAD DE WITH ADDRESS OF 'DESTIN'.
        MVI   BC,LENGTH   ;LOAD BC WITH NUMBER OF BYTES.
LOOP:   MOV   A,M         ;MOVE BYTE FROM SOURCE TO DESTIN.
        STAX  D
        INX   HL          ;INCREMENT POINTER REGISTER HL.
        INX   DE          ;INCREMENT POINTER REGISTER DE.
        DCX   BC          ;DECREMENT COUNTER REGISTER BC.
        MOV   A,B         ;CONTINUE UNTIL BC = Ø.
        ORA   C           ;BC = Ø?
        JNZ   LOOP        ;NO, DO AGAIN.
        RET               ;YES, END.
        DSEG              ;DATA SEGMENT:
SOURCE: DS    LENGTH      ;ALLOCATE 16 BYTES TO SOURCE.
DESTIN: DS    LENGTH      ;ALLOCATE 16 BYTES TO DESTIN.
        END
```

ⓒ
```
        .TITLE MOVE
        .STATIC
SOURCE: .BLKB 16
DESTIN: .BLKB 16
        .ENDSEG
        MOVMB SOURCE,DESTIN,16
```

Bild 2. Quellcodes für einen 16-Byte-Datentransfer im Vergleich.
ⓐ- Pascal-Programm
ⓑ- 8080-Assembler-Programm
ⓒ- NS16000-Assembler-Programm

Registeradressierung:
Der Operand steht in einem Register.

Daneben gibt es noch sechs weitere, speziell zur Unterstützung von Hochsprachen wichtige Adressierungsarten.

Register-relative Adressierung:
Sie entspricht im Grunde der Adressierung durch einen Registerwert, mit dem Unterschied, daß im Befehlswort noch ein Zusatzwert (oft als Offset oder Displacement bezeichnet) enthalten ist, der zum Inhalt des Registers dazuaddiert wird und so die Adresse ergibt (Bild 4).

Zeiger-relative Adressierung:
Wie register-relativ, nur werden jeweils ein Zeigerregister (Stackpointer u. ä.) und ein Zusatzwert zu einer Adresse zusammengestellt.

Speicher-relative Adressierung:
Hier ergibt sich die Operandenadresse aus einem Speicherinhalt und einem Zusatzwert.

'Top of Stack'-Adressierung:
Der Speicherplatz, auf dem sich der Operand befindet, wird durch den Stackpointer adressiert.

'External'-Adressierung:
Ist das Programm aus Modulen aufgebaut, kommt es vor, daß man während der Abarbeitung eines Programms auf die Werte eines anderen Programmes zugreifen muß. Für diesen Zweck gibt es eine Adressierungsart, die sich 'External' nennt.

Skalierter Index:
Zusätzlich zu jeder Adressierungsart, außer zur direkten Adressierung, kann die skalierte Adressierung gewählt werden. Sie spielt bei der Verarbeitung von Bit-Feldern in der Programmiersprache Pascal eine wichtige Rolle.

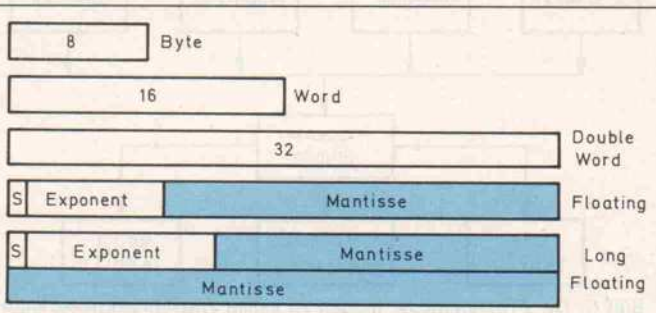


Bild 3. Datentypen beim NS32032

Symmetrie

Die Länge der Ausführungsbefehle kann von einem bis zu 13 Bytes variieren. Ein Standard-Befehlscode ist zum Beispiel wie in Bild 5 aufgebaut.
Symmetrie ist eine Eigenschaft, die es gestattet, daß für jeden Operanden alle neun möglichen Adressierungsarten gelten, zum Beispiel

MOVB R0,A
(Schiebe den Inhalt des Registers R0 auf einen Speicherplatz, der mit A definiert ist. Die Länge der Operanden betrage 1 Byte (B).)
oder:
MOVW HANS,TOS
(Schiebe den Inhalt von HANS in eine Speicherzelle, die mit dem Stackpointer adressiert ist. Die Operanden sollen die Länge 1 Word (W) haben.)

Symmetrie sagt auch aus, daß jeder Operation jeder Datentyp zugeordnet werden kann. Der Befehl MUL (multipliziere) ist zum Beispiel möglich mit:

W	= Word	MULW
B	= Byte	MULB
D	= Double Word	MULD
Q	= Quadword	MULQ
F	= Fließkomma	MULF
L	= Fließkomma mit doppelter Genauigkeit	MULL

Arithmetik

Der Prozessor arbeitet mit Zweioperandenarithmetik. Operand 1 wird mit Operand 2 verknüpft, das Ergebnis wird in Operand 2 geschrieben.
A + B = B ADDW A,B
(Addiere A und B und lege das Ergebnis in B ab.)
R0 + R1 = R1 ADDB R0,R1
(Addiere Register R0 mit Register R1 und lege das Ergebnis in R1 ab.)
Einen 'Akkumulator' gibt es nicht mehr, da jedes Register und jeder Speicherplatz als 'Akkumulator' verwendet werden kann. Wegen der Symmetrie kann man für die Operanden 1 und 2 auch jede Adressierungsart benutzen, der Umweg über den 'Akkumulator' entfällt dann ganz. Diese Eigenschaften sind für die Unterstützung von Hochsprachen sehr wichtig.

NS16000-Befehlssatz

'Call External Procedure', 'Exit', 'Enter' und 'Return' sind Befehle zum Unterpro-

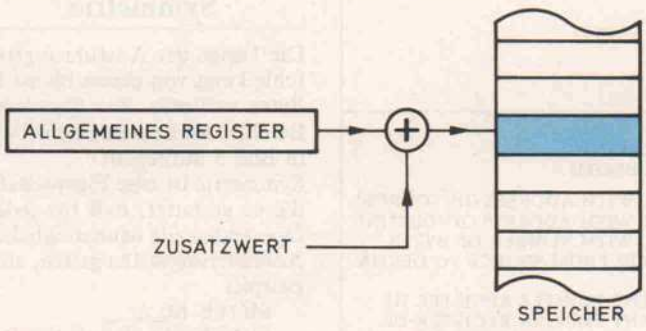


Bild 4. Register-relative Adressierung mit Zusatzwert (Offset)

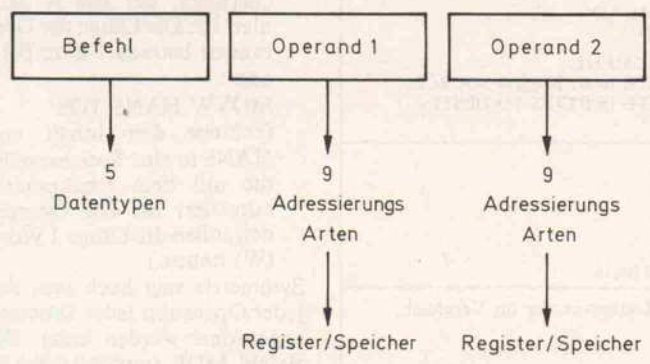


Bild 5. Befehlsstruktur im Maschinencode

grammaufruf. Automatisch retten sie die Register und manipulieren den Stackpointer. Spezielle String-Befehle gestatten die Abarbeitung von Stringfunktionen. Die Register R0 bis R4 übernehmen dabei besondere Aufgaben. Wichtig bei Pascal ist die Behandlung von Feldern und Arrays. Die Prozessoren der NS16000-Serie verfügen auch hierfür über einschlägige Befehle.

Moderne Hochsprachen (Pascal, C) gestatten die Aufteilung der Programme in Module. Jedes Modul kann unabhängig von allen anderen Modulen entwickelt und getestet werden. Beispielsweise kann ein Unterprogramm zur Berechnung von Quadratwurzeln unabhängig vom Hauptprogramm entwickelt und später dann im Hauptprogramm aufgerufen werden.

Der besondere Vorteil davon ist, daß man auch käufliche Pascal- oder C-Programme für seine Zwecke einbinden kann. Die Prozessoren der NS16000-Serie unterstützen als erste einen Programmaufbau in Modulen (Bild 6).

Die fertigen Programme werden von ihrer Quelle in den Befehlscode des NS32032 compiliert. Da, wie schon erwähnt, der Befehlscode an Hochsprachen angepaßt ist, entstehen dadurch kaum aufwendigere Assembler-Programme als unter Verwendung von einfachen Compilern. Quellprogramme in Assemblersprache werden assembliert.

Danach müssen die Programme gebunden (im Englischen spricht man von 'linken') werden, die einzelnen Programmteile werden also in die gewünschten ROM-Bereiche geschrieben. Gleichzeitig werden Tabellen im ROM angelegt, die Auskunft darüber geben, an welcher Stelle der Programmcode der Module beginnt, wo die globalen Variablen untergebracht sind und wie auf die Daten der anderen Programmodule zuzugreifen ist (Linkbasis). Diese Tabellen nennt man Modul-Tabelle (Bild 7) und Link-Tabelle.

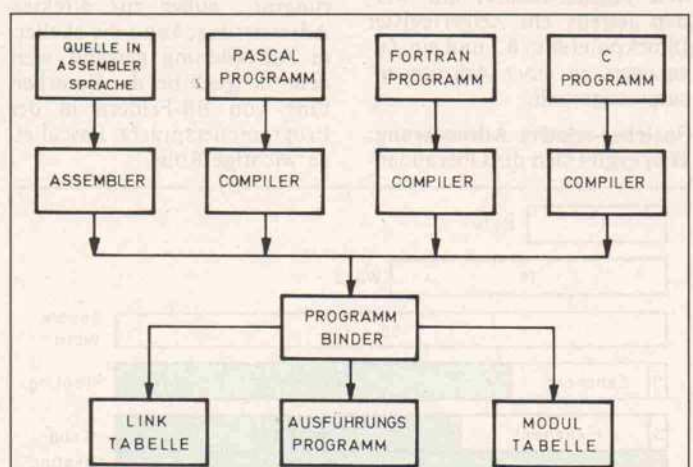


Bild 6. Die Programmteile werden zu einem Ausführungsprogramm gebunden.

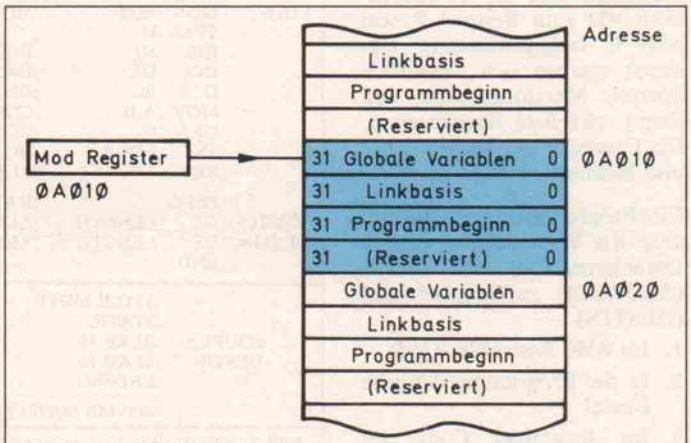


Bild 7. Die Modultabelle

Virtual Memory

24 Adreßleitungen der CPU 32032 legen den Adreßraum auf 16-M-Speicherzellen fest.

Der Optimalzustand wäre, diesen Adreßraum mit direkt zugreifbaren RAM- oder ROM-Speichern voll auszunutzen. Dies ist mit heutigen Speichertechnologien wegen des hohen Kosten- und Platzaufwandes jedoch nicht möglich.

Um aber trotzdem den gesamten Bereich auszunutzen, bedient man sich der virtuellen Speicherung (Bild 8). Wie der Name schon sagt, sieht der Mikrocomputer den Speicher so, als wären alle verfügbaren Adressen auch Speicherplätze, also als läge ein voller 16-M-Adreßbereich vor.

In Wirklichkeit ist der Speicher unterteilt in einen RAM-Bereich, also einen direkt zugreifbaren Halbleiterspeicher, und einen Massenspeicher, also Platte oder Band. Da sowieso nie zur gleichen Zeit der gesamte Adreßbereich benötigt wird, kann man die gerade benötigten Programmteile in einem RAM einlagern und die gerade nicht erforderlichen in einem Massenspeicher ablegen.

Der gesamte virtuelle Speicherbereich wird in Seiten (pages) unterteilt, von denen jede 512 Speicherzellen groß ist. Die gerade benötigten Seiten werden in den RAM-Bereich kopiert, dort eventuell manipuliert und nach längerem Nichtgebrauch wieder zurückgegeben oder verworfen. Benötigt der Mikrocomputer einen Programmteil, der nicht im RAM-Bereich vorhanden ist, wird der gerade ausgeführte Befehl angehalten. Der erforderliche Programmteil wird aus dem Massenspeicher in das RAM geladen. Danach wird der unterbrochene Befehl wiederholt.

Dieses Verfahren nennt man 'Demand Paging', es wird hier erstmals in einem Mikroprozessorsystem angewandt. Großcomputer, wie zum Beispiel die VAX/PDP 11 von der Firma Digital Equipment oder der IBM-370-Rechner, arbeiten mit diesem Verfahren der Speicherverwaltung.

Eine andere Speicherverwaltungstechnik ist die Segmentierung. Hier werden ganze Programme vom Massenspeicher ein- beziehungsweise ausgelagert. Das Mikroprozessorsy-

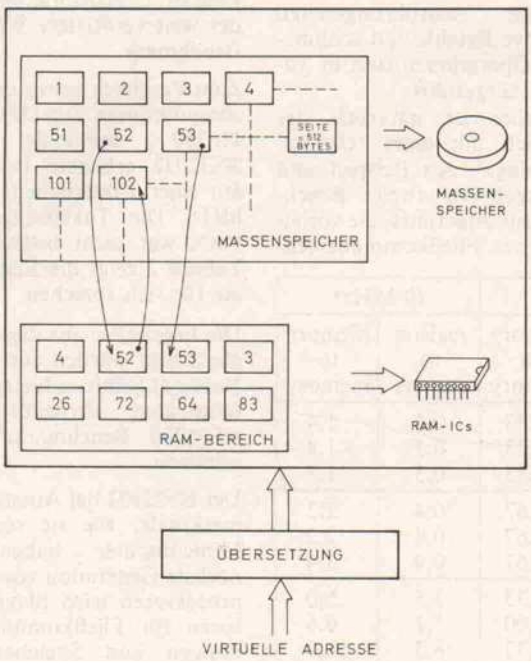


Bild 8. Der virtuelle Speicherraum ist aufgeteilt in RAM- und Massenspeicherbereich.

stem MC 68000 benutzt dieses Prinzip. Erfahrungen aus Großcomputern haben jedoch gezeigt, daß das Verfahren des 'Demand Paging' günstiger ist als die 'Segmentierung', weshalb sich National Semiconductor für das erstere entschied.

Die Verwaltung der Speicher übernimmt die 'Memory Management Unit', kurz MMU (NS16082). Dieser Slaveprozessor sitzt in einem separaten 48-Pin-Gehäuse und ist mit den Adreß-, Daten- und Steuerleitungen der CPU verbunden.

Der Einsatz der MMU bietet weitere Vorteile:

— Die teuren RAM-Bereiche können sehr klein dimensioniert werden. Die billigen Massenspeicher, Platte oder Band, können ein Maximum von Daten aufnehmen. Auf diese Weise hat man die Vorteile von RAMs, nämlich schneller Speicherzugriff, mit denen des Massenspeichers (billig) vereint.

— Jeder Speicherseite kann ein gesondertes Lese- und Schreibrecht zugeordnet werden, so daß ein Schutz vor Benutzereingriffen gegeben ist.

— Der Speicherbereich läßt sich in zwei Räume aufteilen, in einen nur für den Benutzer zugänglichen, und in einen für den Systemverwalter zugreifbaren Teil. Auf diese Weise entsteht Sicherheit vor unbefugtem Verändern von Daten.

Interne Architektur

Die CPU NS32032 ist in einem 68 Pin, 24 mm x 24 mm großen Chip-Carrier-Gehäuse untergebracht. Die Abmessungen des Chips selbst betragen 7 mm x 7 mm (siehe Foto). Zur Stromversorgung genügt eine einfache 5-V-Spannungsquelle.

Der Adreßbus ist 24 Bit und der Datenbus 32 Bit breit. Die Adreß- und Datenleitungen werden von den Anschlüssen AD0 bis AD23 gemultiplext aus der CPU geführt. Die restlichen Datenleitungen D24 bis D31 besitzen eigene Ein-/Ausgänge.

Die CPU besitzt acht 32-Bit-Register R0 bis R7). Sie können zum Halten von Operanden für jeden Befehl verwendet werden. Jedes Register kann auch als 'Akkumulator' benutzt werden. Bei 'String'-Befehlen haben die Register R0 bis R4 besondere Funktionen (Bild 9).

Acht fest zugeordnete Register der CPU speichern Adressen und Zustandsinformationen, die während des Programmablaufes entstehen (Bild 10).

Programmzähler und Statusregister sind wohl hinlänglich bekannt. Das Modulregister beinhaltet den Modulzeiger des gerade ablaufenden Programmmoduls (siehe auch Bild 7). Die CPUs der NS16000-Serie enthalten zwei voneinander unabhängige Stackpointer (SP0 und SP1). SP0 bezeichnet man als Interrupt-Stackpointer. Er wird benutzt, wenn die CPU im Supervisormode (s.u.) ist. Der SP1 oder Anwenderstackpoin-

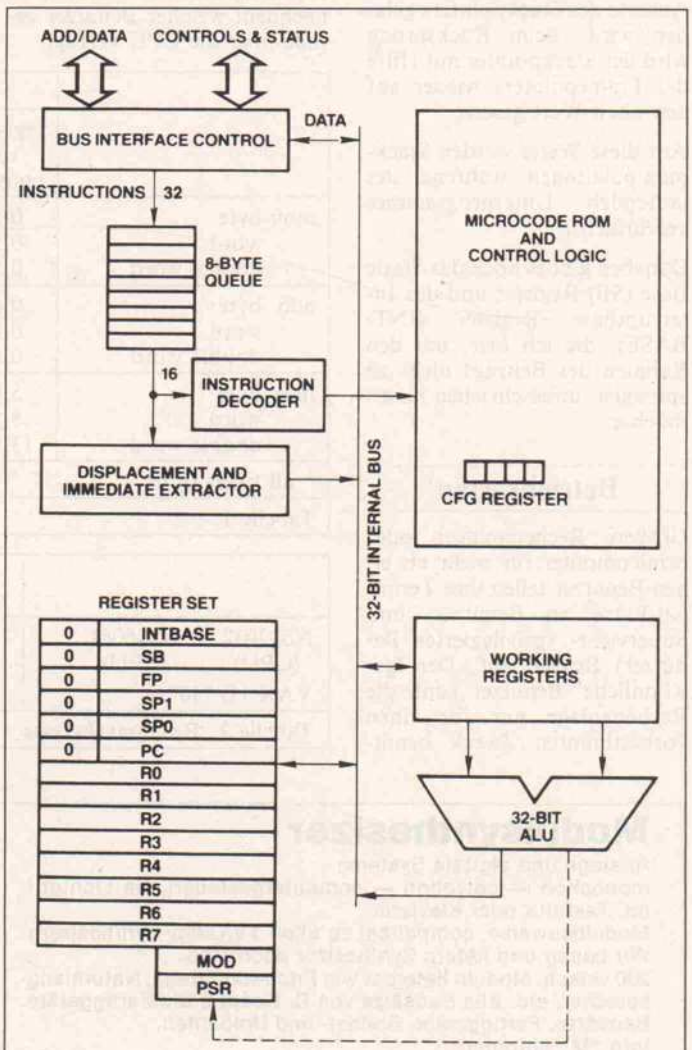


Bild 9. Interne Architektur des 32032

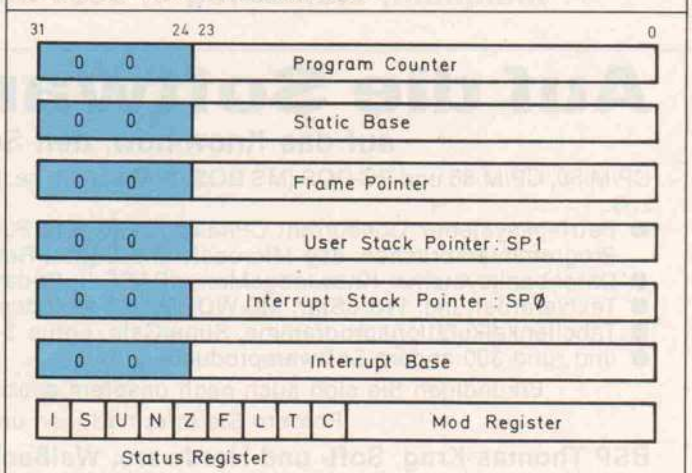


Bild 10. Die fest zugeordneten Register

ter kann vom Programmanwender beispielsweise für 'Top-of-Stack'-Adressierungen verwendet werden. Ob SP0 oder SP1 verwendet wird, hängt davon ab, wie das I-Bit im Zustandsregister gesetzt ist.

Der Framepointer ist ein Register, das nach einem Unterprogrammaufruf mit der aktuellen Adresse des Stackpointers geladen wird. Beim Rücksprung wird der Stackpointer mit Hilfe des Framepointers wieder auf den alten Wert gesetzt.

Auf diese Weise werden Stackmanipulationen während des laufenden Unterprogrammes vereinfacht.

Daneben gibt es noch das Static Base (SB) Register und das Interruptbase Register (INTBASE), die ich hier, um den Rahmen des Beitrags nicht zu sprengen, unbeschrieben lassen möchte.

Betriebsarten

Größere Rechenanlagen oder Minicomputer für mehr als einen Benutzer teilen ihre Terminal-Plätze in Benutzer- und Supervisor- (privilegierter Benutzer) Betrieb auf. Der 'gewöhnliche' Benutzer kann die Rechenanlage nur für einen vorbestimmten Zweck benutzen. Er kann die Programme nicht löschen oder ändern und nur die ihm zugeteilten Programme betreiben.

Der Supervisor hingegen hat die 'Macht', in alle Programme einzugreifen. Die Prozessoren der NS16000-Serie können zwischen diesen beiden Betriebsarten umgeschaltet werden, je nachdem welcher Benutzer gerade über die CPU verfügt.

Rechenleistung

Typische Ausführungszeiten für Move-Befehle und arithmetische Operationen sind in Tabelle 1 aufgeführt.
Deutlicher ist natürlich der Vergleich mit einer 'Großrechenanlage'. Als Beispiel sind die Ergebnisse zweier Benchmarktests angeführt, die vor allem von Fließkommaberechnungen Gebrauch machen. Es sind dies eine 1024-Punkte-Fourier-Transformation und der weit verbreitete Whetstone Benchmark.

Zum Vergleich wurde ein Minicomputer vom Typ DEC VAX-11/750 verwendet. Der NS32032 arbeitete beim Test mit einer Taktfrequenz von 10 MHz. Die Taktfrequenz der VAX war nicht beeinflußbar. Tabelle 2 zeigt die Ergebnisse, die für sich sprechen.

Die Ergebnisse aus diesem Vergleichstest wurden aus der von National Semiconductor veröffentlichten Broschüre 'The NS16000 Benchmarks' übernommen.

Der NS32032 hat Ausstattungsmerkmale, wie sie sonst nur Minicomputer haben. Die nächste Generation von Mikroprozessoren wird Slaveprozessoren für Fließkommaberechnungen und Speicherverwaltung bereits auf dem CPU-Chip mit integriert haben.

Als nächstes will National Semiconductor weitere Slaveprozessoren und Peripheriebausteine zur NS16000-Serie anbieten. Zum Beispiel eine Steuerung für direkten Speicherzugriff, einen Fehlerkorrekturchip und eine Plattenlaufwerksteuerung. □

	6 MHz*		10 MHz*	
	register to register	memory to memory	register to register	memory to memory
mov byte	0,50	2,33	0,3	1,4
word	0,50	2,33	0,3	1,4
double word	0,50	2,83	0,3	1,7
add byte	0,67	3,67	0,4	2,2
word	0,67	3,67	0,4	2,2
double word	0,67	5,67	0,4	3,4
mul byte	5,83	8,33	3,5	5,0
word	8,50	11,00	5,1	6,6
double word	13,83	18,33	8,3	11,0

* all times in μs

Tabelle 1.

	Fourier-transformation	Whetstone
NS32032 + NS16081 (CPU) (FPU)	3,5 μs	7,8 μs
VAX-11/750	4,0 μs	6,0 μs

Tabelle 2. Benchmark-Tests

Modulsynthesizer
Analoge und digitale Systeme
monophon — polyphon — computergesteuert, via Lichtgriffel, Tastatur oder Klaviatur.
Modulbauweise, compatibel zu allen 1 V/Oktav-Synthesizern.
Wir bauen und liefern Synthesizer nach Maß.
200 versch. Moduln lieferbar wie Pitch-to-Voltage, Naturklangspeicher, etc. Alle Bausätze von D. Doepfer als Fertiggeräte.
Bausätze, Fertiggeräte, Sonder- und Umbauten.
Info "M" anfordern.

s/w Graphic-Interface für μP's
Komplett auf Europakarte, Auflösung 256 x 256, (adressierbar und darstellbar), 4 Bildspeicherebenen, (umschaltbar, getrennt für Display bzw. Write) High Intensity Attribut zur Hervorhebung einzelner Objekte oder Buchstaben. Bildspeicher auslesbar, Wort- und Pixelweise, einfacher Cursor-Darstellung mit passendem Adapter für alle PC's und HC's, 'Lightpen und Joy-Stick-Anschluß, BAS-Video-Ausgang (7 MHz—75 Ohm) Graphic-karte komplett mit Befehlssatz.
Info "G" anfordern.

P. Meinhold, Eichenweg 4, 5900 Siegen 1 - Trupbach, Tel. 02 71/3 74 21

Auf die Software kommt es an,
auf das Know-how, den Support und auch den Preis!
CP/M-80, CP/M-86 und PC-DOS (MS DOS)-Software in ca. 50 verschiedenen Diskettenformaten in der Regel ab Lager, z. B.:
- Betriebssysteme: Concurrent CP/M-86, CP/M IBM PC/XT (DM 212,99 incl. MwSt.)
  Programmiersprachen: alle Microsoft und Digital Research-Sprachen
- Datenbanksysteme: KnowledgeMan, dBASE II, Friday!
- Textverarbeitung: WordStar, MS-WORD, in Verbindung mit der MS-MOUSE
- Tabellenkalkulationsprogramme: SuperCalc, Lotus 1-2-3, Multiplan
- und rund 300 andere Softwareprodukte

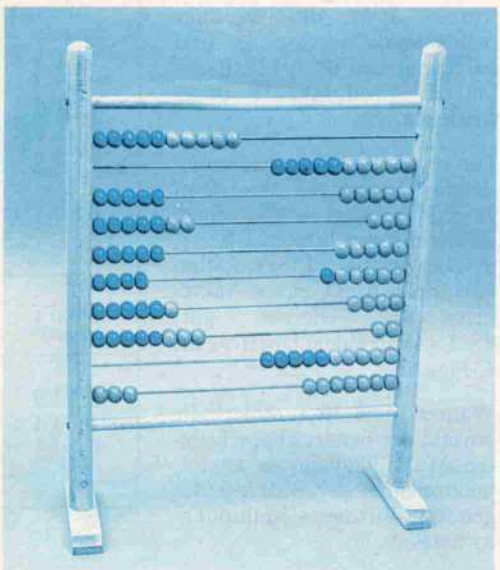
Erkundigen Sie sich auch nach unserem speziellen IBM Personal Computer-Hardwareangebot!
Fordern Sie Informationen und unsere aktuelle Preisliste an:
BSP Thomas Krug, Soft- und Hardware, Weißenburgstr. 49, Postfach 11 03 24, D-8400 Regensburg
Tel. 09 41/5 19 45 und 5 18 66, Tlx 6 52 510

ASSEMBLER

Arithmetik-Unterricht für 6502 und Z 80

Ulrich Schinke

Teil 2:
Multiplizieren und Dividieren



Die Prozessoren 6502 und Z 80 kennen wie fast alle 8-Bit-CPUs keine Befehle zum Multiplizieren und Dividieren. Diese Grundrechenarten müssen durch Programme realisiert werden, die sich im wesentlichen auf Additions- und Subtraktionsbefehle stützen. Im zweiten Teil der 'Nachhilfestunde' für CPUs geht es um die Grundlagen für diese 'Hilfsprogramme', die an fertig programmierten Beispielen gezeigt werden.

Bei der Programmierung von Multiplikations- und Divisionsroutinen sind neben den Additions- und Subtraktionsbefehlen die Rotations- und Schiebebefehle wichtig. Sie bewirken eine Verschiebung der Datenbits um eine Binärstelle. Dadurch erhält jedes Bit als neue Wertigkeit die seines 'Nachbarn' zur Rechten oder Linken. Bei endlicher Wortbreite, normalerweise 1 Byte, 'fällt' dabei an einer Seite des Wortes ein Bit hinaus, welches im Carry-Flag aufgefangen wird. Demzufolge wird auf der anderen Seite des Wortes eine Binärstelle frei.

Die verschiedenen Befehlstypen unterscheiden sich nun lediglich durch die Behandlung des so freiwerdenden Bits:

Bei Verschiebungen erhält dieses eine Ø, bei Rotationen den vorherigen Wert des Carry-Flags (Bild 1).

Somit sind Schiebeoperationen nichts anderes, als Rotationen mit gelöschtem Carry.

Für die 'zirkulare' (8-Bit-) Rotation muß das C-Flag zuvor mit dem Wert des MSB beziehungsweise LSB des zu rotierenden Wortes geladen werden. Die einzelnen Bits dieses Wortes werden hierdurch zwar zyklisch vertauscht, nicht jedoch ihr Wert oder ihre Reihenfolge verändert. Ein 1-Byte-Operand beinhaltet somit nach acht zyklischen Rotationen wieder sein ursprüngliches Bitmuster.

Die Verarbeitung von Mehrbyteoperanden muß durch eine Programmschleife realisiert werden, da sich die Rotations- und Schiebebefehle sowohl beim 6502 als auch beim Z 80 auf eine Wortbreite von 8 Bit (1 Byte) beziehen. Hierbei reicht das C-Flag das herausgeschobene Bit jeweils von Byte zu Byte weiter.

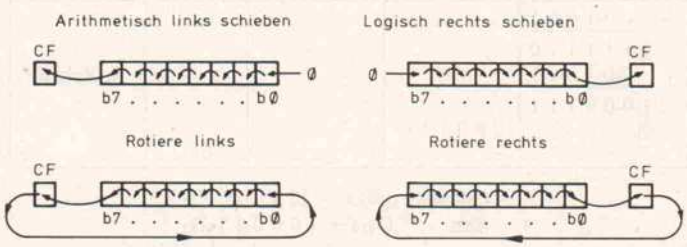


Bild 1. Rotier- und Schiebeoperationen

Fazit:

Mit Hilfe der Rotationsbefehle lassen sich alle Schiebe- und Rotieroperationen für Operanden beliebiger Wortlänge implementieren, indem zuvor das Carry-Flag mit dem entsprechenden Wert geladen wird.

Die in diesem Artikel aufgeführten Programme liefern zahlreiche Beispiele für den Einsatz von Rotationsbefehlen.

Im einfachsten Fall kann eine Multiplikation durch eine fortgesetzte Addition ersetzt werden:

$$7 * 3 = 7 + 7 + 7 = 21.$$

Verständlicherweise wächst die Abarbeitungszeit derartiger Programme mit dem Wert des Multiplikators und ist, besonders für Mehrbyteoperanden, nicht mehr akzeptabel.

Einen Sonderfall stellt die Multiplikation und Division mit der Basis des gewählten Zahlensystems dar. Im Dualzahlensystem entspricht das arithmetische Linksschieben der Multiplikation mit zwei, das logische Rechtsschieben der Division durch zwei.

Leistungsfähigere Programme stützen sich auf die bei der schriftlichen Multiplikation und Division angewendeten Methoden. Die Analyse dieser Rechenvorschriften führt auf die in den Bildern 2 und 3 abgebildeten Flußdiagramme. Die hierzu angegebenen Beispiele (Bild 4 und 5) dienen der Überprüfung und dem besseren Verständnis der Algorithmen. Sie helfen überdies, Fehler bei der Umsetzung der Flußdiagramme in die jeweilige Assemblersprache zu vermeiden.

Bei der Multiplikation (Bild 6) werden zunächst Teilprodukte gebildet und diese dann stellenrichtig addiert. Im Binären entartet die Multiplikation zu einer Ja-/Nein-Entscheidung, ob der Multiplikand zum Zwischenergebnis addiert wird oder nicht.

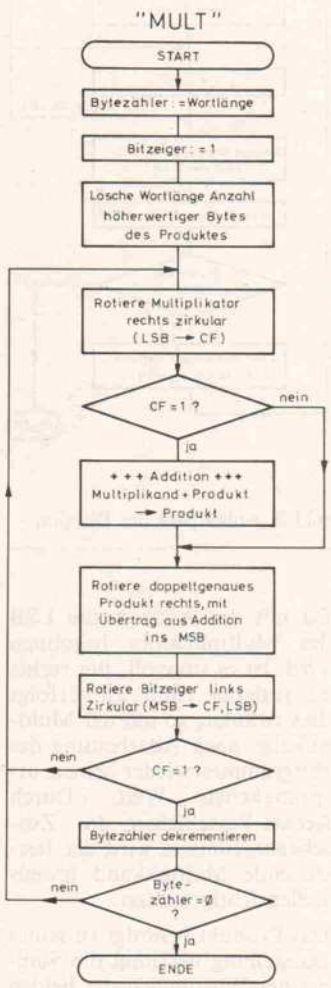


Bild 2. Ablaufplan der Multiplikation

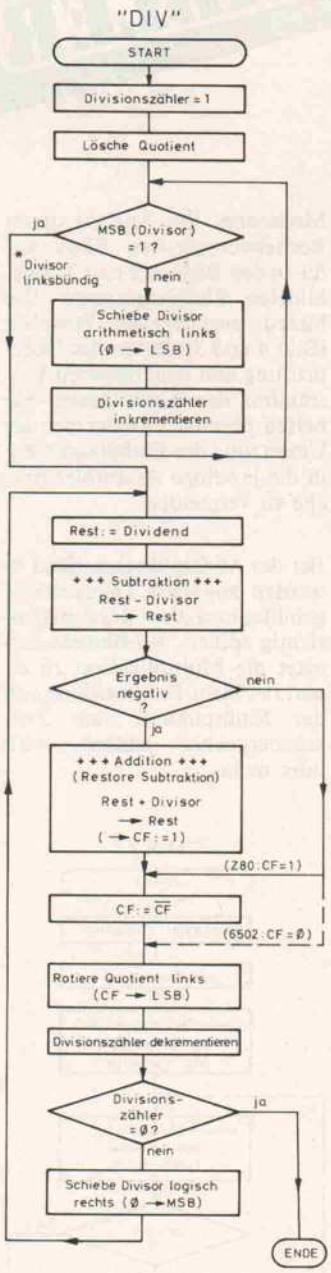


Bild 3: Ablaufplan der Division

Da mit der Abfrage des LSB des Multiplikators begonnen wird, ist es sinnvoll, ihn rechts zu rotieren (Bild 2). Erfolgt dies zirkular, so hat der Multiplikator nach Abarbeitung des Programmes wieder seinen ursprünglichen Wert. Durch Rechts-Verschieben des Zwischenergebnisses wird der feststehende Multiplikand jeweils stellenrichtig addiert.

Das Produkt benötigt zu seiner Darstellung maximal die Summe der Wortlängen der beiden Operanden. Bei zwei gleichlangen Operanden mit je 8 Bit reichen somit in jedem Falle 16 Bit zur Produktdarstellung aus. 'Gerechnet' wird aber nur mit der höherwertigen Produkthälfte, die daher vor Beginn der Multiplikation gelöscht werden muß. Nachdem die Multiplikation vollständig durchlaufen wurde, steht das Produkt rechtsbündig im Speicher und hat damit den alten Inhalt der unteren Produkthälfte überschrieben.

Die etwas ungewöhnlich anmaßende Verwendung eines 'Bitzeigers' realisiert lediglich einen Zähler mit den Werten 1,2,..., 8. Die zirkulare Linksrotation entspricht somit einem Inkrementieren. Hierbei gilt: 8 + 1 = 1 mit einem Übertrag ins C-Flag.

Während der 'Bytezähler' die Anzahl der bereits abgearbeiteten Multiplikatorbytes angibt, kennzeichnet der Bitzeiger das getestete Bit dieses Multiplikatorbytes.

Der Divisionsalgorithmus (Bild 7) ist dem der Multiplikation ähnlich. Der Divisor wird, beginnend beim MSB des Dividenden, jeweils stellenrichtig vom Dividenden subtrahiert. Aus diesem Grund wird der Divisor zunächst linksbündig geschoben. Die Anzahl der hierzu notwendigen Schiebeoperationen plus 1 entspricht der Anzahl auszuführender Subtraktionen. Bei gleicher Wertigkeit des Divisor- und Dividenden-MSB ist somit eine erfolgreiche Subtraktion denkbar.

War der Dividend kleiner als der Divisor, wird das Ergebnis negativ. In diesem Fall erhält der Quotient im LSB eine Ø, sonst eine 1.

Während ein Mensch 'erkennt', welche zweier Zahlen die größere ist, muß der Computer dies testen. Divisionsprogramme arbeiten somit nach dem Prinzip 'Try and Error' (Bild 3). Der Divisor wird in jedem Falle vom Dividenden subtrahiert. Wird hierbei das Ergebnis negativ, so muß die Subtraktion durch eine erneute Addition des Divisors wieder storniert werden (restore method).

Die Z80-CPU (Tabellen 1,2) unterstützt die modulare Programmierung, indem sie Unterprogrammaufrufe (CALL) und Rücksprünge (RET) in Abhängigkeit der Flagzustände gestattet, ähnlich den Verzweigungs-

$14 \cdot 9 = 126$

C_D Multiplikand: $(14)_{10} = (1\ 1\ 1\ 0)_2$
C_D Multiplikator: $(9)_{10} = (1\ 0\ 0\ 1)_2$
Wortbreite := 2 Bits/Byte
Wortlänge := 2 Bytes

x := beliebiges Bit Ø oder 1

Produkt	Multiplikator	Bitzeiger	Bytezähler	
0 0 0 0 X X X X	1 0 0 1	0 1	1 0	
	1 1 0 0			
+ 1 1 1 0				1. Zyklus
0 1 1 1 0 X X X				
0 1 1 1 0 X X X		0 1 0		
0 0 1 1 1 0 X X	0 0 1 1 0			
		1 0 1	0 1	2. Zyklus
0 0 0 1 1 1 0 X	0 0 0 1 1			
		0 1 0		3. Zyklus
	1 1 0 0 1			
+ 1 1 1 0				
0 1 1 1 1 1 1 0 X				
0 1 1 1 1 1 1 0				4. Zyklus
		1 0 1	0 0	

→ Produkt: = $(126)_{10} = (0111\ 1110)_2$

Bild 4: Ausführliches Beispiel zur Multiplikation

$61 : 9 = 6$ Rest 7

→ Dividend : $(61)_{10} = (1\ 1\ .\ 1\ 1\ 0\ 1)_2$
→ Divisor : $(9)_{10} = (0\ 0\ .\ 1\ 0\ 0\ 1)_2$
Wortbreite := 2 Bits/Byte
Wortlänge := 3 Bytes

Divisor	Divzähler	
0 0 1 0 0 1	0 1	Divisor linksbündig
← 0 1 0 0 1 0	1 0	→ Divisionszähler = 3
← 1 0 0 1 0 0	1 1	

Rest: = Dividend

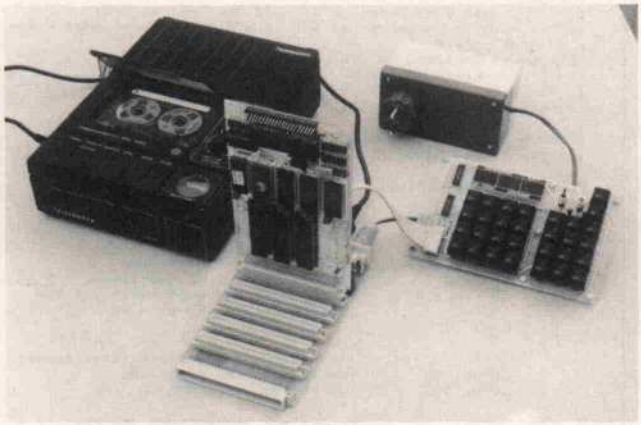
Rest	Quotient	Divisor	Divisionszähler	
1 1 1 1 0 1	0 0 0 0 0 0	1 0 0 1 0 0	1 1	
— 1 0 0 1 0 0				
0 0 1 1 0 0 1				1. Zyklus
1				
	0 0 0 0 0 1			
		→ 0 1 0 0 1 0	1 0	
— 0 1 0 0 1 0				
0 0 0 0 1 1 1				
1				
	0 0 0 0 1 1		0 1	2. Zyklus
		→ 0 0 1 0 0 1		
— 0 1 0 1 0 0 1				
1 1 1 1 1 1 0				
+ 0 1 0 1 0 0 1				3. Zyklus
1 0 0 0 1 1 1				
0	0 0 0 1 1 0		0 0	

→ Quotient : $(6)_{10} = (0\ 0\ 0\ 1\ 1\ 0)_2$
Rest : $(7)_{10} = (0\ 0\ 0\ 1\ 1\ 1)_2$

Bild 5: Ausführliches Beispiel zur Division

COBOLD

IHR Lern- und Proficomputer auf drei Platinen!
Der ideale Einstieg in die Microprozessortechnik



COBOLD — ein Computer mit zauberhaften Qualitäten dank eines neuen, raffinierten Hardware-Konzepts und eines sagenhaft komfortablen Betriebssystems. Auf drei Platinen.
— ein Maschinensprache-Computer auf Basis 6502/65C02, der auch Textverarbeitung, BASIC und FORTH kann.
— der sinnvollste Einstieg in die Microprozessortechnik.
— der Computer für alle — auch Ihre — Problemstellungen.
— beschrieben mit Bauanleitung in ELRAD 3, 4 + 5/83.

Lernen auch Sie zaubern wie ein Cobold — steigen Sie ein in die Microprozessortechnik mit dem neuen elrad-COBOLD-System! Fordern Sie Prospekte an!

Die Komplett-Ausstattungen:

GRUNDVERSION: (CIM 65-Prozessorkarte, Basis- und TD-Platine) mit CPU 6502. RIOT 6532, 2 K RAM, Monitor-EPROM. Basisplatine bestückt mit 1 Federleiste.
Bausatz ..... **DM 298,—**
Bausatz mit fertiger CPU-Karte ..... **DM 389,—**
Fertig aufgebautes System ..... **DM 449,—**

ERWEITERTE VERSION (Grundversion mit 4 K RAM, 3x RIOT 6532, Basisplatine mit 5 Federleisten).
Bausatz ..... **DM 398,—**
Bausatz mit fertiger CPU-Karte ..... **DM 498,—**
Fertig aufgebautes System ..... **DM 549,—**

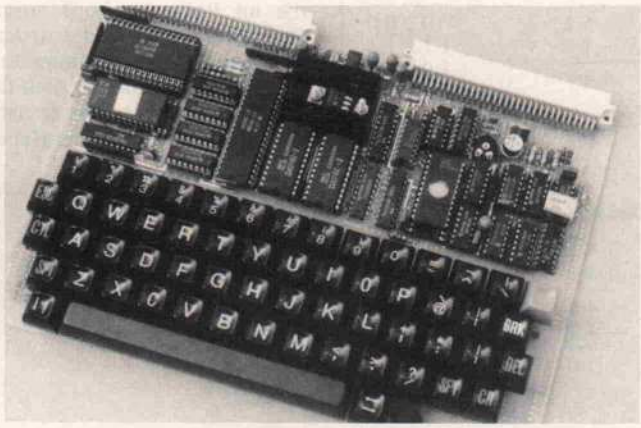
NETZTEIL für den COBOLD im Steckergehäuse **DM 49,—** (Bausatz) bzw. **DM 86,—** (fertig) (auch für c't-Terminal geeignet).

DAS HANDBUCH für den COBOLD: „6502/65C02 Maschinensprache" von C. Persson **DM 48,—**.

Intelligentes Terminal
mit professionellen Attributen:

c't-Terminal

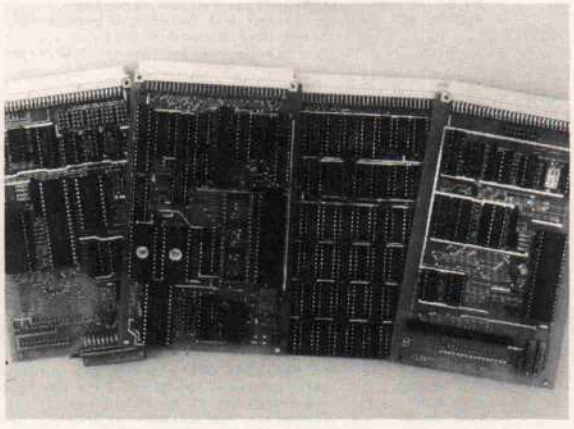
DER Terminal-Computer auf Doppel-Euro-Karte
mit oder ohne integrierter Tastatur!



— beschrieben in c't Nr. 12/83 und 1/84
— 6511-Singlechipcomputer mit 6545-Videocontroller
— 4 KB-Bildwiederholspeicher (scrollbar)
— Bildformat 80 x 25 oder 64 x 20 (per Software umschaltbar)
— Zeichenmatrix 8 x 11 (bei 80 x 25) oder 8 x 13 (bei 64 x 20)
— max. 8 Zeichensätze (inkl. Blockgrafik)
— Invers-, Blink-Modus, Breitschrift, halbe Helligkeit
— serielles Interface (V 24- oder TTL-Pegel)
— integrierte Centronics-Schnittstelle
— integrierte Spannungsregelung und -wandlung für V24
— 8-bit-parallel (ASCII) oder 8 x 9 Tastenmatrix Tastaturanschluß
— PREISE: **Version A** (ohne Tastatur)
Bausatz **DM 449,—; DM 549,—** Fertigkarte
Platinenmaße 233 x 85 mm
Version B (mit integrierter Tastatur)
Bausatz **498,—; DM 639,—** Fertigkarte
Platinenmaße 233 x 160 mm
Prospektmaterial auf Anforderung!

Endlich: c't-86

Das 1. echte 16-bit-Microcomputer-System
der Welt (auch) zum Selbstbau!



Ein Vier-Karten-System, basierend auf dem für 16 bit erweiterten ECB-Bus:
— echte 16-bit-Rechenleistung
— kein neuer, sondern ein weitverbreiteter Bus
— dadurch bereits existierende ECB-Peripherie-Karten einsetzbar
— Betriebssysteme CP/M-86 und MS-DOS II
— **Komplettpreis unter DM 1 700,— (Bausatz)**
— vorgestellt in Heft 1, 2 + 3/84 von c't — dem neuen Magazin für Computertechnik

Die vier Karten:

Platine 1: **CPU-KARTE mit 8086,** optional 8087 Arithmetik-Prozessor, 8259 Interrupt-Controller, 8 KB Monitorprogramm mit CP/M-86-Urlader.
Komplett-Bausatz ..... **DM 349,—; DM 449,—** Fertigkarte

Platine 2: **I/O-KARTE** mit V-24-Interface für Terminal-Anschluß, Centronics-Schnittstelle, Kassettenrekorder-Interface und Timer.
Komplett-Bausatz ..... **DM 249,—; DM 349,—** Fertigkarte

Platine 3: **FLOPPY-CONTROLLER-KARTE** zum Anschluß bis zu 4 Laufwerken 5¼ oder 8 Zoll (auch gemischt) mit dem neuen Controller-IC WD 2797.
Komplett-Bausatz ..... **DM 498,—; DM 598,—** Fertigkarte

Platine 4: **256-KB-RAM-KARTE** mit 128 oder 256 KB dyn. RAM (max. 3 Karten einsetzbar ≙ 768 KB RAM!).
Komplett-Bausatz **DM 598,—** (128 KB) bzw. **DM 899,—** (256 KB), **DM 698,—** bzw. **DM 999,—** Fertigkarte

Geplante Ergänzungen: CPU-Karte mit 68000, SASI-Interface, Vollgrafikkarte, Z 80 Subprozessor-Karte.

Bus-Karte mit 10 Steckplätzen — fertig — ..... **DM 169,—**
Leerplatinen, Floppy-Laufwerke, Netzteile und Gehäuse ..... **auf Anfrage**

Komplettsystem, anschlußfähig im Gehäuse, mit Laufwerken ab DM 3985,—

Fordern Sie Prospekte an!

VERSAND: per NN (+ Versandkosten) oder per Vorauskasse (V-Scheck oder Überweisung auf Pschkto Han 1429 28-308, keine Versandkosten). Ausland nur gegen Vorauszahlung (+ DM 15,— Versandkosten). Alle Preise inkl. MwSt. **Händleranfragen erwünscht.**

MARFLOW — COMPUTING GmbH

Brüderstraße 2 · 3000 Hannover 1 · Telefon 05 11/32 60 98

```
13 · 14     1101 · 1110
-------     -----------
  52            0000
  13           1101
-------       1101
 182         1101
            ---
             111
          --------
          10110110
```

Bild 6: Multiplikation

```
 182 : 14 = 13          10110110 : 1110 = 1101
-14                    -1110
---                    -----
 42                     01000 01
-42                    -1110
---                    -----
  0                     0001110
                       -  1110
                       ------
                            0

 191 : 15 = 12 Rest 11   10111111 : 1111 = 1100
-15                    -1111                 t(12)10
---                    -----
 41                     01000 01
 30                    -1111
---                    -----
 11                     0001011        ← Rest 11
```

Bild 7: Division ohne und mit Rest

anderen Programme beliebig oft zugreifen, müssen jedoch die im Programmkopf aufgeführten Eingaben bereitstellen. Bei jedem Unterprogrammaufruf muß der Prozessor sich die Stelle 'merken', an die er nach Abarbeitung der aufgerufenen Routine zurückkehren muß. Hierzu legt die CPU die Rückkehradresse auf einen Stapel (Stack). Dieser Speicher wird nach dem 'LIFO'-Prinzip (Last In, First Out) verwaltet, da das den Verzweigungsbefehlen (JR, JMP). Der Vorteil dieser Programmiertechnik liegt in der größeren Übersichtlichkeit und sicheren Testbarkeit. Auf diese Unterprogramme können alle

Tabelle 1.

```
;Aufgabe:"MULT" multipliziert zwei Mehrbytewörter mit der in
;       : WORTL festzulegenden Genauigkeit und speichert das
;       : Produkt mit doppelter Wortlänge ab PRD.
;       : Eingaben werden durch das Programm nicht zerstört.
;Eingaben : Multiplikand ab MLD
;         : Multiplikator ab MTR
;         : Wortlänge in WORTL
;Ausgaben : Produkt ab PRD mit 2*(WORTL)
;Variablen: PRD, MLD, MTR, WORTL, BIT2, BITZ+1
;Unterprogramme: ADD, RRW
;Zerstört: AF, BC, HL, DE, IX

MULT:   LD  A,(WORTL)   ;Wortlänge über Akku
        LD  C,A         ;ins C-Register laden
        LD  IX,BITZ     ;Bitzeiger (BITZ): nächstes abzuar-
        LD  (IX+0),1    ;beitendes Bit eines Multiplikatorbytes
        LD  (IX+1),C    ;Bytezähler (BITZ+1): noch abzuarbeitende
                        ;Bytes des Multiplikators

        LD  HL,PRD      ;+++ Produkt löschen +++
        LD  B,C         ;Wortzeiger --> HL, Wortlänge --> B
CLR1:   LD  (HL),0      ;Obere Hälfte des Produktes löschen, da
        INC HL          ;unter Hälfte ohnehin durch Schiebeopera-
        DJNZ  CLR1      ;tionen überschrieben wird

                        ;+ Multiplikator rechts zirkular rotieren
MLTI:   LD HL,MTR       ;Wortzeiger --> HL und
        PUSH HL         ;  auf Stack zwischenspeichern
        ADD  HL,BC      ;LB-Adresse (Multiplikator) =
        DEC HL          ;  Wortzeiger + Wortlänge - 1
        RRC (HL)        ;LSB (Multiplikator)
        RL  (HL)        ;     --> CF kopieren
        POP HL          ;Wortzeiger wieder vom Stack holen
        CALL RRW        ;Multiplikator  1 Bit rechts zirkular
                        ;rotieren

                        ;+++ Additionseingaben :
        LD DE,PRD       ;Wortzeiger (Zwischenergebnis) --> DE
        LD HL,MLD       ;Wortzeiger (Operand) --> HL

                        ;Falls LSB (Multiplikand) 1 war (-> CF=1):
        CALL C,ADD      ;  Multiplikand + Produkt --> Produkt

                        ;++ Produkt rechts rotieren ++
        EX  DE,HL       ;Wortzeiger (Produkt) vom DE ins HL
        CALL RRW        ;Rotiere doppelt genaüs Produkt um 1 Bit
        CALL RRW        ;rechts, mit übertrag aus Addition ins MSB

        RLC (IX+0)      ;nächstes Bit des Multiplikatorbytes
        JR NC,MLTI      ;Wiederhole bis Bit 7 abgearbeitet wurde
        DEC (IX+1)      ;8 Bits abgearbeitet --> Bytzähler - 1
        JR NZ,MLTI      ;Wiederhole bis alle Bytes abgearbeitet

        RET
```

Tabelle 2.

```
;Aufgabe:"DIV" teilt 2 Mehrbytewörter durch einander, speichert
;       : den Quotienten ab QTT und den Rest ab DVD.
;       : Alle Wörter haben die in WORTL festzulengende Genau-
;       : igkeit.
;Eingaben: Dividend ab DVD
;         : Divisor ab DVS
;         : Wortlänge in WORTL
;Ausgaben: Quotient ab QTT
;         : Rest ab DVD
;Unterpramme: RRW, RLW, ADD, SUB
;Zerstört: Dividend, HL, BC, DE, AF, IX

DIV:    LD  A,(WORTL)   ;Wortlänge über Akku
        LD  C,A         ;ins C-Register laden
        LD  IX,DIVZ     ;Divisionszähler (DIVZ) := 1, da
        LD  (IX+0),1    ;wenigstens 1 Division möglich

        LD  HL,QTT      ;+++ Quotient löschen +++
        LD  B,C         ;Wortzeiger --> HL, Wortlänge --> B
CLR2:   LD  (HL),0      ;Alle Bytes des Quotienten auf 00H
        INC HL          ;setzen, beginnend beim HB
        DJNZ  CLR2

                        ;++ Additions-, Subtraktionseingaben :
        LD DE,DVD       ;Wortzeiger (Rest) --> DE
        LD HL,DVS       ;Wortzeiger (Divisor) --> HL

                        ;+ Divisor linksbündig schieben ++

RL1:    RLC (HL)        ;Kopiere MSB (Divisor) ins CF
        RR (HL)         ;Falls CF=1 , ist Divisor links-
        JR C,DVIS       ;bündig --> weiter bei DVIS

        XOR A           ;CF := 0  --> Schiebe Divisor 1 Bit
        CALL RLW        ;arithmetisch links
        INC (IX+0)      ;Inkrementiere Divisionszähler
        JR RL1          ;Verzweige zur nächsten MSB Abfrage

DVIS:   CALL SUB        ;< Rest - Divisor --> Rest >
                        ;Falls Ergebnis negativ (CF=1)
                        ; --> Restore Subtraktion :
        CALL C,ADD      ; < Rest + Divisor --> Rest >

                        ;++ Rotiere Quotient links ++
        PUSH HL         ;Rette HL auf Stack
        CCF             ;CF := CF'
        LD HL,QTT       ;Rotiere Quotient 1 Bit links,
        CALL RLW        ;  mit CF' ins LSB
        POP HL          ;Hole HL wieder vom Stack

        DEC (IX+0)      ;Dekrementiere Divisionszähler

        RET Z           ;Fertig, falls Divisionszähler=0 !

                        ;++ Schiebe Divisor rechts logisch ++
        PUSH HL         ;Rette HL auf Stack
        XOR A           ;CF := 0  --> Schiebe Divisor 1 Bit
        CALL RRW        ;rechts logisch
        POP HL          ;Hole HL wieder vom Stack

        JR DVIS         ;Verzweige zur nächsten Division
```

Unterprogramm

```
;Aufgabe:"ADD" addiert einen Operanden zu einem Zwischenergebnis
;       : mit der im BC-Register festzulegenden Genauigkeit.
;Eingaben: Wortlänge -- BC
;         : Wortzeiger (Operand) -- HL
;         : Wortzeiger (Zwischenergebnis) -- DE
;Ausgaben: Ergebnis ab (DE)
;Zerstört: AF

ADD:    EX DE,HL        ;Inhalt des DE- mit dem des HL-Registers
        ADD HL,BC       ;vertauschen, um mit 2 Byte Addition die
        EX DE,HL        ;LB-Adressen +1 zu berechnen,
        ADD HL,BC       ;zunächst vom Zwischenergebnis (DE),
                        ;dann vom Operanden (HL)
        LD B,C          ;Wortlänge ins B-Register kopieren
                        ;
        XOR A           ;Akku(A) und C-Flag löschen
NADD:   DEC DE          ;nächsthöheres Byte adressieren
        DEC HL          ;+++ Byteweise Addititon +++
        LD A,(DE)       ;(Zwischenergebnis + Operand + mög-
        ADC A,(HL)      ; licher übertrag) --> Zwischenergebnis
        LD (DE),A       ; + Wortlänge (B) - 1
        DJNZ NADD       ;Wiederhole bis alle Bytes abgearbeitet
        RET


;Aufgabe:"SUB" subtrahiert einen Operanden von einem Zwischener-
;       : gebnis mit der im BC-Register festzulegenden Genauigkei
;Eingaben: Wortlänge -- BC
;         : Wortzeiger (Operand) -- HL
;         : Wortzeiger (Zwischenergebnis) -- DE
;Ausgaben: Ergebnis ab (DE)
;Zerstört: AF

SUB:    EX DE,HL        ;Inhalt des DE- mit dem des HL-Registers
        ADD HL,BC       ;vertauschen, um mit 2 Byte Addition die
        EX DE,HL        ;LB-Adressen +1 zu berechnen,
        ADD HL,BC       ;zunächst vom Zwischenergebnis (DE),
                        ;dann vom Operanden (HL)
        LD B,C          ;Wortlänge ins B-Register kopieren
        XOR A           ;Akku(A) und C-Flag löschen

NSUB:   DEC DE          ;nächsthöheres Byte adressieren
        DEC HL          ;+++ Byteweise Subtraktion +++
        LD A,(DE)       ;(Zwischenergebnis - Operand - mög-
        SBC A,(HL)      ; licher übertrag) --> Zwischenergebnis
        LD (DE),A       ; + Wortlänge (B) - 1
        DJNZ NSUB       ;Wiederhole bis alle Bytes abgearbeitet
        RET
```

Unterprogramm

```
;Aufgabe:"RRW" rotiert ein Wort um 1 Bit rechts, wobei die
;       : Operationsart durch (CF) bestimmt wird.
;Eingaben: Wortzeiger -- HL, Wortlänge -- C
;       : CF: 0   --> Schiebe rechts logisch
;       :     MSB --> Schiebe rechts arithmetisch
;       :     LSB --> Rotiere rechts zirkular
;Ausgaben: 1 Bit rechts verschobenes Wort ab Wortzeiger (HL)
;       : (HL) = Wortzeiger + Wortlänge
;Zerstört: Flags, HL

RRW:    LD B,C          ;Wortlänge ins B-Reg. kopieren
RR:     RR (HL)         ;Wort byteweise um 1 Bit rechts rotie-
        INC HL          ;ren, mit CF ins MSB
        DJNZ RR         ;Wiederhole bis alle Bytes abgearbeitet
        RET


;Aufgabe:"RLW" rotiert ein Wort um 1 Bit links, wobei die
;       : Operationsart durch (CF) bestimmt wird.
;Eingaben: Wortzeiger -- HL, Wortlänge -- BC
;       : CF: 0   --> Schiebe links arithmetisch
;       :     MSB --> Rotiere links zirkular
;Ausgaben: 1 Bit links verschobenes Wort ab Wortzeiger (HL)
;Zerstört: Flags

RLW:    PUSH AF         ;Flags auf Stack zwischenspeichern
        ADD HL,BC       ;Wortzeiger + Wortlänge --> HL
        POP AF          ;Flags wieder vom Stack holen

        LD B,C          ;Wortlänge ins B-Reg. kopieren
RL:     DEC HL          ;Wort byteweise um 1 Bit links rotie-
        RL (HL)         ;ren, mit CF ins LSB
        DJNZ RL         ;Wiederhole bis alle Bytes abgearbeitet
        RET

        END
```

zuletzt aufgestapelte Byte auch als erstes wieder heruntergenommen werden muß.

Die 'Höhe' des Stapels bestimmt der Stapelzeiger (Stackpointer), der immer die Adresse des nächsten freien Stapelplatzes enthält.

Der Stack kann auch vom Anwender zum Zwischenspeichern von Variablen sowie zum Kopieren und Austauchen von Speicher- und Registerinhalten mit Hilfe der PUSH- und POP-/PULL-Befehle genutzt werden. Es ist jedoch darauf zu achten, die von der CPU angelegte Ordnung nicht zu stören. Alle aufgestapelten Daten müssen wieder vom Stack 'heruntergenommen' worden sein, bevor die CPU durch einen 'Return'-Befehl veranlaßt wird, ins Hauptprogramm zurückzuspringen. Sie interpretiert nämlich die beiden zu oberst auf dem Stack liegenden Daten als Rücksprungadresse, unabhängig davon, ob dies eventuell dort 'vergessene' Operanden sein mögen. Solche Fehler führen im allgemeinen zum 'Absturz' des Rechners.

Die Abfrage des Divisors-MSB wird durch zwei Rotierbefehle (RRC (HL) und RR (HL)) realisiert, die das MSB in das C-Flag kopieren. Dies demonstriert eine weitere Anwendung der Rotationsoperationen.

Der 6502 (Tabellen 3,4) unterstützt gegenüber dem Z 80 die Einbindung der abzuarbeitenden Sequenzen in das Hauptprogramm. Hierdurch wird meist eine kürzere Rechenzeit erzielt, da keine Parameter an Unterprogramme übergeben werden müssen und die Programm-Module speziell angepaßt werden können.

Durch die 'indirekte Adressierung', die tatsächliche Adresse wird aus der Summe einer Basisadresse plus dem Inhalt des Distanzregisters (X) bestimmt, sind für die einzelnen Programmblöcke gegenüber den Z 80-Programmen weniger Befehle notwendig.

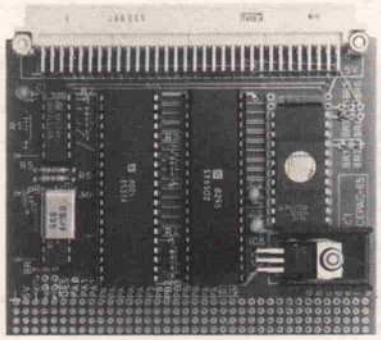
Obwohl der 6502 einen absoluten Sprungbefehl besitzt, sollten innerhalb eines überschaubaren Programmblocks die schnelleren Branchbefehle diesem vorgezogen werden, wobei die Verzweigung sich auf ein mit Sicherheit gesetztes oder gelöschtes Flag stützt.

Besondere Beachtung muß die unterschiedliche Behandlung des Carry-Flags bei der Subtraktion finden:

Während beim Z 80 das C-Flag bei einem negativen Ergebnis gesetzt (CF = 1) und bei positivem Resultat gelöscht wird (CF = 0), handhabt der 6502 dies genau umgekehrt. Im Divisionsprogramm des Z 80 (Programm 2) wird die zur Stornierung notwendige Addition nur dann ausgeführt, wenn das C-Flag gesetzt ist; im 6502-Programm (Programm 4) wird sie bei gesetztem Carry übersprungen.

Wer Programmierung nicht als Selbstzweck betreibt, steht spätestens jetzt vor der Aufgabe, sich die hier entwickelten Programme 'MULT' und 'DIV'

NEU: CEPAC-65



Der preisgünstige Single-Board-Computer für Festprogramm-Anwendungen. Bausatz — Version A — ..... **DM 69,—**
Version B ..... **DM 89,—**

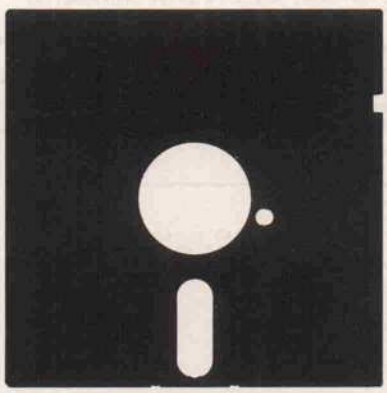
Massenweise MONITORE! Massenweise

Heath-Zenith ZVM 122 (bernstein) ..... **DM 355,—**
(12 Zoll, 15 MHz) ZVM 123 (grün) ..... **DM 355,—**
PRINCE (bernstein) ..... **DM 555,—**
(12 Zoll, 25 MHz) (grün) ..... **DM 495,—**

LOW-COST-DRUCKER
EUROPRINT 6311 (100 Z/s) ..... **DM 999,—**

TANDON-FLOPPY-Laufwerke
— u. a. von IBM für gut befunden ...

TM 50-1 (ss/dd, 250 KB, 1 x 40 Track) ..... **DM 586,—**
TM 50-2 (ds/dd, 500 KB, 2 x 40 Track) ..... **DM 698,—**
TM 55-2 (ds/dd, 500 KB, 2 x 40 Track) ..... **DM 798,—**
TM 55-4 (ds/gd, 1 MB, 2 x 80 Track) ..... **DM 948,—**
TM 55-2 + 55-4 sind mikroprozessorgesteuert (extrem genaue Kopfpositionierung!)
Alle Laufwerke in Slimline-Ausführung (½ Bauhöhe).
P.S. Tandon hat einen Weltmarktanteil von 75 %!



Mit uns speichern Sie richtig!

5,25"	10 Stck.
SS/SD 48TPI	52,—
SS/DD 48TPI	58,—
SS/QD 96TPI	74,—
DS/DD 48TPI	79,50
DS/QD 96TPI	84,—
8"	
SS/SD	58,—
SS/DD	67,—
DS/DD	79,50

Größere Mengen Preis auf Anfrage. Preise pro Stück inkl. MwSt. zzgl. Versandkostenanteil: DM 5,—.

Leerplatinen für c't-Projekte:
c't-Terminal — Version A — (ohne Tastatur) ..... **DM 59,—**
c't-Terminal — Version B — (mit Tastatur) ..... **DM 75,—**
Universelles Netzteil ..... **DM 14,—**
c't-86-System: Prozessor-Karte ..... **DM 85,—**
RAM-Karte ..... **DM 88,—**
I/O-Karte ..... **DM 69,—**
Floppy-Karte ..... **DM 65,—**
BUS-Platine (96pol.) für 10 Steckplätze ..... **DM 49,—**
c't-Sprachsynthesizer ..... **DM 21,—**
CEPAC-65 ..... auf Anfrage

VERSAND: per NN (+ Versandkosten) oder per Vorauskasse (V-Scheck oder Überweisung auf Pschkto Han 1429 28-308, keine Versandkosten). Ausland nur gegen Vorauszahlung (+ DM 15,— Versandkosten). Alle Preise inkl. MwSt.

MARFLOW-COMPUTING GmbH
Brüderstraße 2 · 3000 Hannover 1 · Telefon 05 11/32 60 98

Tabelle 3.

```
;Aufgabe:"MULT" multipliziert zwei Mehrbytewörter mit der in
;       : WORTL fest zulegenden Genauigkeit und speichert das
;       : Produkt mit doppelter Wortlänge ab PRD.
;       : Eingaben werden durch das Programm nicht zerstört.
;Eingaben : Multiplikand ab MLD
;         : Multiplikator ab MTR
;         : Wortlänge in WORTL
;Ausgaben : Produkt ab PRD mit 2*(WORTL)
;Variablen: PRD, MLD, MTR, WORTL, BITZ, BYTEZ
;Zerstört: alle Register

MULT:   CLD             ;Binärmode !
        LDA WORTL       ;Wortlänge über Akku
        TAX             ;X- Zählregister kopieren
        DEX             ;Wortzähler (X) := Wortlänge -1
        STA BYTEZ       ;Bytezähler (BYTEZ) := Wortlänge
        ASL             ;Wortlänge * 2 -->
        STA DWRTL       ;  Doppelte Wortlänge (DWRTL)
        LDA #1          ;Bitzeiger (BITZ) := 1
        STA BITZ        ;

        LDA #0          ;+++ Produkt löschen +++
CLR1:   STA PRD,X       ;Obere Hälfte des Produktes löschen,
        DEX             ;da untere Hälfte ohnehin durch Schiebe-
                        ;operationen überschrieben wird
        BPL CLR1        ;Wiederhole, bis Wortlänge Bytes
                        ;gelöscht sind

MLTI:   LDX WORTL       ;+ Multiplikator rechts zirkular rotieren +
        DEX             ;Wortzähler := Wortlänge -1
        LDA MTR,X       ;LSB (Multiplikators) über Akku
        ROR             ;ins CF kopieren
        LDX #0          ;Wortzähler (X) := 0

        LDY WORTL       ;Wortzähler (Y) := Wortlänge
RR1:    ROR MTR,X       ;Multiplikator 1 Bit rechts zirkular
        INX             ;rotieren, beginnend beim HB
        DEY             ;Wiederhole bis alle Bytes abgearbeitet
        BNE RR1         ;wurden --> Wortzähler (Y) = 0

        BCC NOADD       ;überspringe die Addition, falls
                        ;MSB (Multiplikators) 0 war (--> CF = 1)

        DEX             ;Wortzähler (X) := Wortlänge - 1
        CLC             ;CF := 0

ADD:    LDA PRD,X       ;(Zwischenergebnis + Operand + möglicher
        ADC MLD,X       ; übertrag ) --> Zwischenergebnis +
        STA PRD,X       ; Wortlänge - 1 ,beginnend beim LB
        DEX             ;Wiederhole, bis alle Bytes abgearbeitet
        BPL ADD         ; --> Wortzähler (X) < 0

                        ;++ Produkt rechts rotieren ++
NOADD:  LDX #0          ;Rotiere doppelt genaues Produkt um
        LDY WORTL       ;(DWORTL) ins Y-Register kopieren
RR2:    ROR PRD,X       ;1 Bit rechts, mit übertrag aus Addition
        INX             ;ins MSB , beginnend beim HB
        DEY             ;Wiederhole, bis alle Bytes abgearbeitet
        BNE RR2         ; --> Doppelte Wortlänge (DWRTL) = 0

        ROL BITZ        ;Rotiere Bitzähler links
        BCC MLTI        ;Wiederhole, bis Bit 7 abgearbeitet
        ROL BITZ        ;Kopiere CF ins Bit 0
        DEC BYTEZ       ;8 Bits abgearbeitet --> Bytezähler - 1
        BNE MLTI        ;Wiederhole, bis alle Bytes abgearbeitet
                        ; --> Bytezähler = 0 !
        RTS
```

Tabelle 4.

```
;Aufgabe:"DIV" teilt 2 Mehrbytewörter durch einander, speichert
;       : den Quotienten ab QTT und den Rest ab DVD.
;       : Alle Wörter haben die in WORTL festzulengende Genau-
;       : igkeit.
;Eingaben: Dividend ab DVD
;         : Divisor ab DVS
;         : Wortlänge in WORTL
;Ausgaben: Quotient ab QTT
;         : Rest ab DVD
;Variablen: DVD, DVS, QTT, DIVZ
;Zerstört: alle Register

DIV:    CLD             ;Binärmode !
        LDA WORTL       ;Wortlänge über Akku ins
        TAX             ;X- Zählregister kopieren
        DEX             ;Wortzähler (X) := Wortlänge - 1
        LDA #1          ;Divisionszähler (DIVZ) := 1,
        STA DIVZ        ;da wenigstens 1 Division möglich

        LDA #0          ;+++ Quotient löschen +++
CLR2:   STA QTT,X       ;Alle Bytes des Quotienten löschen,
        DEX             ;beginnend beim LB
        BPL CLR2        ;Wiederhole, bis alle Bytes gelöscht

                        ;++ Divisor linksbündig schieben ++
LDIV:   LDA DVS         ;Kopiere HB (Divisors) in Akku
        BMI DVIS        ;linksbündig, falls MSB = 1
        LDX WORTL       ;
        DEX             ;Wortzähler (X) := Wortlänge -1
        CLC             ;CF := 0
RL1:    ROL DVS,X       ;Schiebe Divisor 1 Bit arithmetisch
        DEX             ;links
        BPL RL1         ;Wiederhole, bis alle Bytes verschoben
        INC DIVZ        ;Inkrementiere Divisionszähler
        BCC LDIV        ;Da CF immer 0 --> Verzweige unbedingt
                        ;zu erneuter MSB Abfrage

DVIS:   LDX WORTL       ;
        DEX             ;Wortzähler (X) := Wortlänge - 1
        SEC             ;CF := 1
```

Unterprogramm

```
                        ;++ Byteweise Subtraktion ++
SUB1:   LDA DVD,X       ;(Zwischenergebnis - Operand - möglicher
        SBC DVS,X       ; übertrag ) --> Zwischenergebnis +
        STA DVD,X       ; Wortlänge - 1 ,beginnend beim LB
        DEX             ;Wiederhole, bis alle Bytes abgearbeitet
        BPL SUB1        ; --> Wortzähler (Y) < 0

        BCS NOADD       ;Falls Ergebnis pos (CF=1), überspringe
                        ;Addition
        LDX WORTL       ;Kopiere Wortlänge - 1 in Wortzähler (X)
        DEX             ;

                        ;++ Byteweise Addition ++
ADD2:   LDA DVD,X       ;(Zwischenergebnis + Operand + möglicher
        ADC DVS,X       ; übertrag ) --> Zwischenergebnis +
        STA DVD,X       ; Wortlänge - 1 ,beginnend beim LB
        DEX             ;Wiederhole, bis alle Bytes abgearbeitet
        BPL ADD2        ; --> Wortzähler (Y) < 0
        CLC             ;CF := 0
```

Unterprogramm

```
NOADD:  LDX WORTL       ;Wortlänge - 1 in Wortzeiger (X)
        DEX             ;
                        ;++ Quotient links rotieren ++
RL3:    ROL QTT,X       ;Rotiere Quotient 1 Bit links,
        DEX             ;  mit CF ins LSB
        BPL RL3         ;Wiederhole, bis alle Bytes rotiert

        DEC DIVZ        ;Dekrementiere Divisionszähler
        BNE NEXT        ;Fertig, falls Divisionszähler = 0
        RTS

                        ;++ Divisor rechts logisch schieben ++
NEXT:   LDY WORTL       ;Wortlänge in Wortzähler (Y) kopieren
        LDX #0          ;Wortzähler (X) :=0
        CLC             ;CF:= 0
RR3:    ROR DVS,X       ;Divisor 1 Bit logisch
        INX             ;rechts schieben
        DEY             ;
        BNE RR3         ;Wiederhole, bis alle Bytes rotiert

        BCC DVIS        ;Da CF immer 0 --> Verzweige unbedingt
                        ;zu nächster Division
```

für ihren eigentlichen Zweck, der Multiplikation und Division von Mehrbyteoperanden dienbar zu machen.

Vor der Assemblierung der Programme muß der Anwender den symbolischen Variablen tatsächliche, physikalische Adressen zuweisen. Dies kann durch manuelles Ersetzen der Symbole durch ihre Adressen im Quelltext oder mit Hilfe von Pseudoassembleranweisungen geschehen, zum Beispiel MLD EQU $2000, die diesen Vorgang automatisch während der Assemblierung durchführen.

Alle Routinen können als Subroutinen von einem übergeordneten Programm aufgerufen werden. Sie verarbeiten alle im Programmkopf als Eingaben deklarierten Operanden, die, wie in Bild 8 dargestellt, zuvor im Speicher abgelegt sein müssen. Es ist darauf zu achten, daß der Stapelzeiger einen genügend großen Speicherbereich als Stack verwalten kann. Vereinbarungsgemäß adressiert er diesen vom höchstwertigsten Byte aus abwärts.

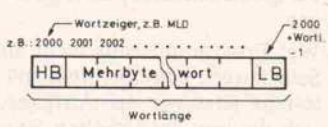


Bild 8: Datenformat

Zum Schluß noch einige Bemerkungen, inwieweit diese Divisions- und Multiplikationsprogramme mit der Zahlendarstellung im Zweierkomplement vereinbar sind: Bei einer im Zweierkomplement dargestellten Zahl repräsentiert das MSB vereinbarungsgemäß das Vorzeichen (Ø——> positiv, 1——> negativ), wobei alle darzustellenden Zahlen die gleiche Wortlänge haben müssen (c't 3/84).

'MULT' ordnet jedoch dem Produkt die doppelte Wortlänge der Operanden zu. Trotzdem ist diese Routine für im Zweierkomplement dargestellte Zahlen nutzbar, wobei dann nur die untere Produkthälfte das Ergebnis repräsentiert.

Demgegenüber setzt 'DIV' positive Integerzahlen voraus, da der Algorithmus das C-Flag als Überlaufanzeige verwendet. Diese Aufgabe übernimmt aber bei der Zweierkomplement-Darstellung das V-Flag! Dies bedeutet nun nicht zwangsläufig eine Umänderung des Divisionsprogrammes, sondern läßt sich durch betragsmäßige Verarbeitung der Zahlen umgehen. Hierbei muß der Benutzer entscheiden, ob Quotient oder Rest negativ ist, und diese dann gegebenenfalls komplementieren. □

SYSTECH GMBH

OLYMPIA-COMPACT: Qualität mit Rechneranschluß

Typenradschreibmaschine mit Universal-Interface (c't-Februar-Heft), kann parallele oder serielle Daten verarbeiten (nur DIL-Schalter umschalten und Stecker austauschen!). Die **Halbzeilenschaltung** erlaubt die **Darstellung mathematischer Formeln (Matrizen, Potenzen).** Typenräder mit verschiedenen Schriften und Formelzeichen sind verfügbar.

Technische Daten:

- — parallele Schnittstelle (kann z. B. anstelle des MX-80 angeschlossen werden)
- — serielle Schnittstelle mit 1200 Baud (paßt z. B. an den Osborne- oder Zenith-Rechner) mit DTR-Handshake (DTR-Pegel wählbar)
- — deutscher Zeichensatz, Datenpuffer mit 44 Zeichen
- — Schreibgeschwindigkeit max. 14 Zeichen/s, 9 Zeichen/s inkl. Warenrückkauf

NEU: Interface-Version 3.2 mit wesentlichen Verbesserungen

Baugruppe mit paralleler Schnittstelle (jetzt auch Acknowledge-Signal verfügbar), RS-232-Schnittstelle mit DTR-Handshake und wählbarem CBM-Zeichensatz (z. B. C-64, VC-20). Wahlweise RS-232-Stecker bzw. Stecker für parallele Schnittstelle, SYCOM 3.2B ..... **DM 348,—**

wie vor, anstelle des CBM-Zeichensatzes XON-XOFF-Protokoll für die RS-232-Schnittstelle, SYCOM 4.0B ..... **DM 348,—**

CBM-Benutzer (2000-, 3000-, 8000er-Serie):
wie vor, für den CBM-IEC-BUS mit CBM-Zeichensatz SYCOM 1.0B ... **DM 398,—**

wie vor, jedoch als Bausatz SYCOM 3.2BS, SYCOM 4.0BS ..... **DM 298,—**

wie vor, jedoch nur Platine & CPU SYCOM 3.2P, SYCOM 4.0P ..... **DM 198,—**

Anschlußleitungen für C-64, VC-20 mit aktivem Interface ..... **DM 69,—**

Anschlußleitungen für Zenith Z 89/90, Osborne, Acorn ..... **DM 49,—**

Es liegen allen Baugruppen und Bausätzen Einbauhinweise, eine Bedienungsanleitung und der Zeichensatz bei.

OLYMPIA-COMPACT Typenradschreibmaschine, komplett mit Interface nach Wahl (SYCOM 3.2BS, SYCOM 4.0BS) ..... **DM 1398,—**
mit Interface für CBM-IEC-BUS, SYCOM 1.0 ..... **DM 1448,—**

c't-Sprachsynthesizer:

Sprachausgabe aus jedem Rechner mit einer parallelen Schnittstelle. Freier Textaufbau durch Phonemsynthese, notwendige Datenrate ca. 10 Byte/s. Programmierbar über BASIC, PASCAL, C oder Assembler. Die Sprachausgabe nach dem c't-Projekt aus dem Februar-Heft.

Sonderausführungen für RS-232 und CBM-IEC-Bus sind lieferbar.

Fertigkarte (parallele Eingabe), Anschluß an VC-20, C-64, CBM 2001..., (über den USER-Port) und jede parallele Druckerschnittstelle möglich,
entsprechende Beschreibung wird mitgeliefert: .. **DM 298,—**

Wie vor, Bausatz: ..... **DM 248,—**

Wie vor, jedoch nur Sprachsynthesizer IC und Platine: ..... **DM 172,—**

Komplettsystem im Gehäuse mit Netztrafo, anschlußfertig: ..... **DM 448,—**

Wie vor, für CBM-IEC-BUS (z. B. CBM 2001,...) oder RS-232-Schnittstelle, anschlußfertig: ..... **DM 548,—**

Einkartenrechner mit Z 80-CPU und Monitor

Für Meß- und Regelsysteme, Steuerungen.

- — ZE-2: 2 Z 80 PIO's, Z 80-Interrupt (IM 2), bis 8K-EPROM, 3K-RAM; 2,5 MHz Takt: ..... **DM 298,—**
- — ZE-64: wie ZE-2. zus. Z 80A CTC, bis 16K-EPROM, 64K-RAM, 4 MHz Takt: ..... **DM 598,—**
- — AD 12/8: passender 8-Kanal-12-Bit-Umsetzer (50 µs/Kanal): ..... **DM 998,—**
- — AT-2: Anzeige und 24 Tasten passend zu ZE-2 & 64: ..... **DM 168,—**
- — ZE-2-SER: Serielle & parallele Schnittstelle für ZE-2 & ZE-64: ..... **DM 98,—**
- — SYS-2/64: Monitorprogramm für ZE-2/64, unterstützt ADU, Schnittstelle, Anzeige und Fehlersuche. Mit ZE-2/64 Hardwarebeschreibung, Monitorbeschreibung, **Listing** und Monitor im 2716-EPROM: ..... **DM 98,—**

Ab April lieferbar:

Datenerfassung mit dem VC-20/C-64
Grundeinheit mit Eingangsbuffer für maximal 8 A/D-Umsetzer, Datenerfassungsprogramm auf Kassette.
Die Grundeinheit wird an den USER-Port angeschlossen. ..... **DM 98,—**

Analog-Digitalumsetzer für den VC-20/C-64
Eingänge für Temperaturfühler (LM 390), ±1 V, ±1 mA sind vorhanden, Auflösung 2000 Schritte, eine Messung pro Sekunde, maximale Entfernung vom Rechner 50 m, 3adrige Zuleitung ..... **DM 98,—**

Referenzspannungsquelle

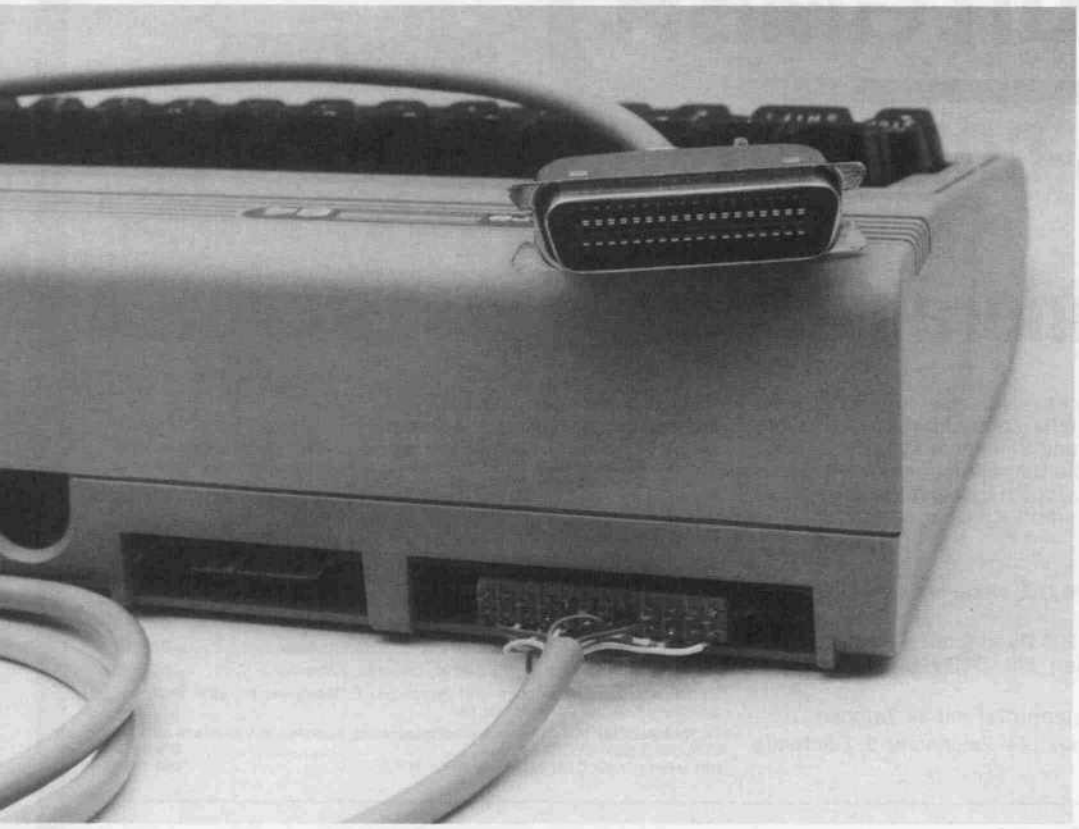
Digital einstellbare Gleichspannungsquelle 10 V in 10000 Schritten zu 1 mV, kurzschlußfest. Restwelligkeit unter 100 Mikrovolt.

Ideal zum Abgleich und zur Kontrolle von Analog-Digital-Umsetzern und als preiswerte Referenzquelle am Arbeitsplatz ..... **DM 398,—**

Auftragsentwicklungen: analoge und digitale Schaltungen.
Mikroprozessoren: Z 80, 8085, MCS-48 und 6801/68701.
Speicherprogrammierte Steuerungen.

Alle Preise einschließlich Mehrwertsteuer, Versand per Nachnahme oder Vorkasse. Lieferzeiten 2...4 Wochen.

SYSTECH GmbH, Gliesmaroder Str. 26, 3300 Braunschweig, Tel.: 05 31/34 56 41 & 37 23 91



Centronics-Schnittstelle für VC-20 und C64

Nur das Kabel löten

Georg Werner

Alle hardwareseitigen Voraussetzungen für den Anschluß eines handelsüblichen Druckers mit Centronics-Schnittstelle sind sowohl beim VC-20 als auch beim C64 erfüllt. Warum Commodore die Software für diese Schnittstelle nicht mitliefert, wird wohl für immer ein Rätsel bleiben. Die paar Bytes, die nötig sind, um die Schnittstelle zu treiben, wären gewiß noch frei gewesen. Vielleicht wollte man die Kompatibilität zu den Commodore-Druckern nicht gefährden ...? — Welche Motive auch immer hinter dieser Produktpolitik stecken mögen, Schwamm drüber! Zum Anschluß eines Druckers Ihrer Wahl brauchen Sie nur ein Verbindungskabel und die hier vorgestellte Software. Auch der c't-Sprachsynthesizer läßt sich an dieser Schnittstelle betreiben.

Zur Erinnerung: Die Centronics-Schnittstelle ist eine Parallel-Schnittstelle mit acht Daten- und mehreren Handshake-Leitungen. Letztere übernehmen die Steuerung des Datenverkehrs ('Handshake': wörtlich 'Händeschütteln'). Der Sender signalisiert mit einem Strobe-Impuls ($\overline{STB}$), daß gültige Daten vorhanden sind und vom Empfänger übernommen werden sollen. Dieser quittiert den Empfang mit einem Acknowledge-Impuls ($\overline{ACK}$). Daneben gibt es eine weitere Handshake-Leitung, nämlich BUSY. Der Empfänger setzt diese Leitung auf log. 1, wenn er ein Datenbyte empfangen hat. Ist er bereit, das nächste Byte zu übernehmen, legt er den Anschluß wieder auf 0-Pegel. Der sendende Computer kann diese Leitung abfragen, um zu erfahren, ob er ein weiteres Byte übertragen kann.

Einen genauen Überblick über die zeitliche Abfolge gibt Bild 1. Wer es noch genauer wissen will, sollte sich den Grundlagen-Artikel 'Parallel-Schnittstellen' in der c't-Ausgabe 1/84 (Seite 74) noch einmal durchlesen.

Bild 2 zeigt die Anschlußbelegung einer Centronics-Schnittstelle. Außer der Daten- und Handshake-Leitung sind noch weitere Steuerleitungen vorhanden. Sie dienen dazu, den angeschlossenen Drucker zu initialisieren und weitergehende Informationen über Fehlerzustände an den Computer zu übertragen. Für den Datentransport werden sie nicht benötigt, denn jeder Fehlerzustand wird vom Drucker zusätzlich durch das Busy-Signal angezeigt. Die zusätzlichen Leitungen werden deshalb in unserem Vorschlag nicht berücksichtigt.

Pin	Signal	Pin	Signal
1	$\overline{\text{Strobe}}$	19	GND
2	Data 0	20	GND
3	Data 1	21	GND
4	Data 2	22	GND
5	Data 3	23	GND
6	Data 4	24	GND
7	Data 5	25	GND
8	Data 6	26	GND
9	Data 7	27	GND
10	$\overline{\text{Ack}}$	28	GND
11	Busy	29	GND
12	Paper out	30	GND
13	NC	31	$\overline{\text{Initial}}$
14	GND	32	$\overline{\text{Error}}$
15	GND	33	GND
16	GND	34	$\overline{\text{CLK}}$
17	Chassis	35	Test
18	+5 V	36	+5 V

Bild 2. Pinbelegung einer Centronics-Schnittstelle

VC-20 und C64 stellen an ihren User-Ports jeweils 24 Leitungen zur Verfügung, von denen

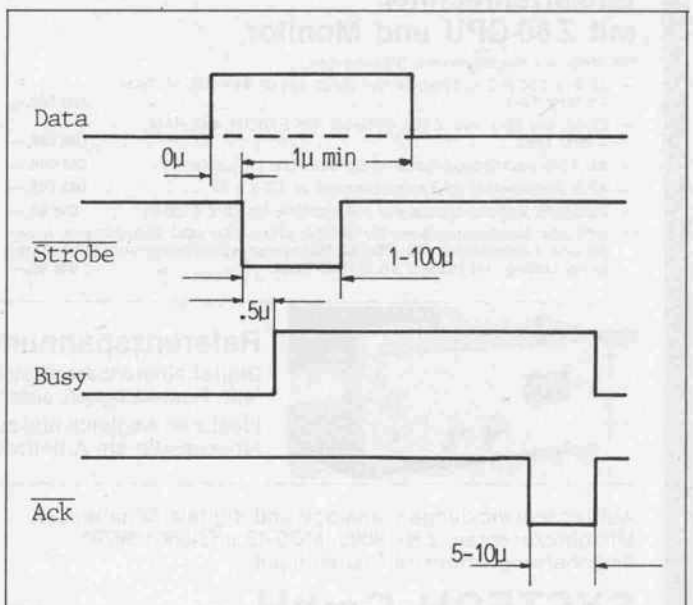


Bild 1. Zeitablauf des Handshakes bei einer Centronics-Schnittstelle

zehn für die Druckerschnittstelle verwendbar sind. Einige dieser Leitungen werden übrigens auch für die RS-232-Karte benötigt. Man muß sich also entscheiden: Die Datenausgabe über den User-Port kann entweder nur seriell (RS-232) oder parallel (Centronics) erfolgen.

Beide Commodore-Computer ähneln sich weitgehend, so daß die folgende Beschreibung im Prinzip auf VC-20 und C 64 anwendbar ist. Wo es Unterschiede gibt, werden diese vermerkt, oder die für den C 64 geltenden Angaben werden in Klammern gesetzt.

Als Datenleitungen werden die Anschlüsse PB0...PB7 verwendet (siehe Bild 3). Beim VC-20 kommen diese direkt aus dem Interfacebaustein VIA 6522 mit der Basisadresse 9110h. Beim C 64 gehören diese Portleitungen zu dem Interface-Chip CIA 6526 mit der Basisadresse DD00h.

Die beiden übrigen Leitungen dienen dazu, den Handshake auszuführen: Nach einer Schreib- oder Leseoperation an Port B liefert CB2 (PC2) einen 0-Impuls, der bei der Centronics-Schnittstelle als Strobe-Signal Verwendung findet. Der Anschluß CB1 (FLAG2) reagiert auf eine negative Flanke des Eingangssignals. Der Interfacebaustein setzt dann ein Flag in seinem Interruptregister. Hier wird das Busy-Signal des Druckers angeschlossen.

Der gesamte Hardware-Aufwand für den Einbau der Centronics-Schnittstelle beschränkt sich also auf die Herstellung eines Verbindungskabels. Bild 3 zeigt, wie der Amphenolstecker der Centronics-Schnittstelle mit dem Stecker für den User-Port zu verbinden ist. Das zwölfadrige Kabel sollte allerdings nicht länger sein als 1,5 Meter, denn die Interface-Bausteine können eine längere Leitung nicht störungsfrei treiben.

Amphenolstecker	Userport
1	M (VC-20) 8 (C 64)
2	C
3	D
4	E
5	F
6	H
7	J
8	K
9	L
11	B
19...29 verbinden	A,N

Bild 3. Verbindung des Amphenolsteckers der Centronics-Schnittstelle mit dem Userport

Drucken in BASIC

Ein einfaches BASIC-Programm genügt, um einen Drucker, den c't-Sprachsynthesizer oder andere Peripheriegeräte zu steuern. Die Struktur eines solchen Programms ist aus dem Beispiel (siehe Tabelle 1) leicht ersichtlich. Zunächst muß die Schnittstelle initialisiert werden. Dafür sorgt das Unterprogramm ab Zeile 20000. In Zeile 20010 werden die Portleitungen des Interface-Bausteins als Ausgänge programmiert. In Zeile 20020 wird der richtige Handshake-Modus vereinbart. Der C 64 kennt nur eine Art von Handshake, deshalb kann für diesen Rechner Zeile 20020 entfallen. In Zeile 20030 wird als erstes Zeichen eine '00' (ASCII: NUL) übertragen. Dieses Zeichen dient dazu, den Drucker zu initialisieren. Die Übertragung ist auch dann notwendig, wenn das Peripheriegerät diesen Befehl nicht versteht, denn es antwortet auf jeden Fall mit einem $\overline{\text{ACK}}$-Impuls. Das ist nötig, damit der Rechner nicht später vor der Ausgabe des ersten Zeichens 'hängenbleibt' (Zeile 10050).

```
10 gosub 20000 :rem  interface initialisieren
20 p$="Drucken mit Centronics"
25 p$ = p$ + chr$(13)
30 gosub 10000 :rem drucken
40 input p$
50 if p$="ende" then end
55 p$ = p$ + chr$(13)
60 gosub 10000 : goto 40
70 :
10000 rem ausgabe des strings
10010 for i = 1 to len(p$)
10020 p = asc(p$)
10030 gosub 15000
10040 p$ = mid$(p$,2)
10050 wait 37136+13,16 :rem c64: 56576+13,16
10060 poke 37136+0,p   :rem c64: 56576+1,p
10070 next
10080 return
10090 :
15000 rem veraendern des zeichensatzes
15010 if (peek(36864+5) and 2) = 0 then return
15015 rem c64: if <peek<53248+24> and 2> = 0 then return
15020 if p < 64 then return
15030 if p > 191 then p = p and 127 :return
15040 p = p or 32 :return
15050 :
20000 rem initialisieren
20010 poke 37136+2,255 :rem c64: 56576+3,255
20020 poke 37136+12,175:rem c64: entfaellt
20030 poke 37136+0,0   :rem c64: 56576+1,0
20040 return

ready.
```

Tabelle 1. BASIC-Programm zur Ansteuerung der Centronics-Schnittstelle

Der zweite Teil ist die eigentliche Druckroutine (10000). Hier wird der String in einzelne Zeichen zerlegt. Diese werden in den ASCII-Code umgesetzt und dann über den Port B ausgegeben. Dabei wartet das Programm stets in Zeile 10050, bis der Drucker den Empfang des zuletzt gesendeten Zeichens mit Acknowledge bestätigt und damit die Bereitschaft zum Empfang des nächsten gemeldet hat.

VC-20 und C 64 können mit verschiedenen Zeichensätzen arbeiten, nämlich mit Großbuchstaben/Grafik und mit Groß- und Kleinschreibung. Im zweiten Zeichensatz wird die ASCII-Norm teilweise nicht eingehalten. Damit dies bei der Ausgabe auf einem Standard-Drucker nicht zu unverständlichem Kauderwelsch führt, wird der Code im Programmsegment ab Zeile 15000 entsprechend umgesetzt. Welche der Betriebsarten der Benutzer ausgewählt hat, erkennt das Programm in Zeile 15010 an der Adresse des Zeichengenerators im Videocontroller. Eine genaue Beschreibung der Codewandlung enthielt der Bericht 'Seriell drucken' in c't 3/84.

Schneller drucken

Für die Textverarbeitung mit einem Schönschriftdrucker oder für die Sprachausgabe mit dem c't-Sprachsynthesizer ist die Geschwindigkeit des BASIC-Programms ausreichend. Wer jedoch einen Matrixdrucker oder einen Plotter betreiben will, wird ein schnelles Maschinenprogramm vorziehen.

Das hier vorgestellte Maschinenprogramm besteht aus drei Teilen:

Die Subroutine CHROUT wird vom Betriebssystem über den Vektor 0326h aufgerufen, wenn ein Zeichen ausgegeben werden soll. Der Computer prüft in diesem Programmsegment, ob die Ausgabe über die Centronics-Schnittstelle erfolgen soll. Ist das nicht der Fall, erfolgt ein Rücksprung in das Betriebssystem. Im anderen Fall wird eine Routine ausgeführt, die ein Zeichen über den Userport ausgibt. Dieser Teil entspricht in seiner Struktur exakt dem BASIC-Programm ab Zeile 10000.

Der zweite Teil ist das Unterprogramm CHKOUT. Dieses Unterprogramm wird vom Betriebssystem über den Vektor 0320h aufgerufen, um festzustellen, auf welchem Gerät eine Ausgabe erfolgen soll. Dabei wird aus der Speicherzelle BAh die Gerätenummer geholt und mit der Nummer der Centronics-Schnittstelle (5) verglichen. Bei Übereinstimmung wird die Nummer in der Speicherzelle 9Ah als die des aktuellen Ausgabegeräts abgelegt. Andernfalls erfolgt wiederum ein Rücksprung in das Betriebssystem.

Der dritte Teil ist die Routine INI. Sie übernimmt die Funktion des BASIC-Programms ab Zeile 20000 und zusätzlich das Setzen der Vektoren 0320h und 0326h. Außerdem wird in diesem Programmteil der Zeiger 'End of RAM' heruntergesetzt, um ein Überschreiben des Maschinenprogramms durch BASIC-Variablen zu verhindern.

Die Routine INI muß vor der ersten Benutzung der Centronics-Schnittstelle mit 'SYS

```
     Assembler "CENTRONICS"

     Achtung für C64 Werte in Klammern
     verwenden!!!


     1D00 48        PHA             ,BEGIN  CHROUT
     1D01 A5 9A     LDA  GERÄT
     1D03 C9 05     CMP  #CENTRON
     1D05 F0 03     BEQ  AUS
     1D07 4C 7D F2  JMP  RÜCK
          (4C CD F1)
AUS  1D0A AD 1D 91  LDA  FLAG
          (AD 0D DD)
     1D0D 29 10     AND  #10
     1D0F F0 F9     BEQ  AUS        ;ENTSPRICHT WAIT
     1D11 AD 05 90  LDA  VIDEO      ;ZEICHEN ÄNDERN
          (AD 18 D0)
     1D14 29 02     AND  #02
     1D16 F0 10     BEQ  OK1
     1D18 68        PLA
     1D19 C9 40     CMP  #40
     1D1B 90 0A     BCC  OK2
     1D1D C9 C0     CMP  #C0
     1D1F 90 04     BCC  CNG
     1D21 29 7F     AND  #7F
     1D23 D0 02     BNE  OK2
     1D25 09 20     ORA  #20
OK2  1D27 48        PHA             ;ENDE ÄNDERN
OK1  1D28 68        PLA
     1D29 8D 10 91  STA  PORT       ;AUSGABE
          (8D 01 DD)
     1D2C 18        CLC
     1D2D 60        RTS
     1D2E EA        NOP             ;BEGIN  CHKOUT
     1D2F 20 CF F3  JSR  FILENR
          (20 0F F3)
     1D32 F0 03     BEQ  1D37
     1D34 4C 84 F7  JMP  FILE NOT OPEN
          (4C 01 F7)
     1D37 20 DF F3  JSR  SETPARA
          (20 1F F3)
     1D3A A5 BA     LDA  GERÄTNR
     1D3C C9 05     CMP  #CENTRON
     1D3E F0 05     BEQ  CEN
     1D40 A5 BA     LDA  GERÄTNR
     1D42 4C 16 F3  JMP  RÜCK1
          (4C 5D F2)
CEN  1D45 4C 2E F3  JMP  RÜCK2
          (4C 75 F2)
     1D48 00        BRK
     1D49 00        BRK
     1D4A 00        BRK
     1D4B 00        BRK
     1D4C A9 FF     LDA  #FF        ;BEGIN INI
     1D4E 8D 12 91  STA  DDRB
          (8D 03 DD)
     1D51 A9 AF     LDA  #AF
          (EA EA)
     1D53 8D 1C 91  STA  INTER
          (EA EA EA)
     1D56 A9 00     LDA  #00
     1D58 8D 10 91  STA  PORT
          (8D 01 DD)
     1D5B 8D 26 03  STA  CHRVEKL
     1D5E A9 **     LDA  #START
     1D60 8D 27 03  STA  CHRVEKH
     1D63 8D 21 03  STA  CHKVEKH
     1D66 85 38     STA  END BASIC
     1D68 A9 2F     LDA  #STRTCHK
     1D6A 8D 20 03  STA  CHKVEKL
     1D6D 60        RTS
```

Tabelle 2. ASSEMBLER-Programm 'Centronics'

(INI)' aufgerufen werden. Dasselbe gilt nach Betätigen der 'Restore'-Taste, denn dabei werden vom Betriebssystem sämtliche Vektoren zurückgesetzt und die Portbausteine neu initialisiert. Vergißt man diesen Befehl, erscheint bei einem Druckversuch ein 'device not present error'.

	START	Anfangsadresse
VC-20 8K	3F	3FØØ
VC-20 16K	5F	5FØØ
C64	9F	9FØØ

Tabelle 3. Anfangsadressen und Werte der Variablen START bei verschiedenem Speicherausbau des Rechners

Das Maschinenprogramm ist so geschrieben, daß es an jeder Stelle im Speicher des VC-20 und des C64 arbeiten kann. Dadurch ergibt sich eine einfache Anpassung an die jeweilige Rechnerkonfiguration. Nur ein einziges Byte muß in Abhängigkeit vom Startpunkt eingesetzt werden. Dieses Byte ist mit '**' gekennzeichnet und wird im folgenden mit 'START' bezeichnet.

Bei VC-20 mit maximal 3 KByte Speichererweiterung kann man das Maschinenprogramm am einfachsten im RAM direkt unterhalb des Bildschirmspeichers ablegen. START erhält dann den Wert 1Dh. Beim C64 und beim VC-20 mit 8- oder 16-K-Erweiterung sollte das Programm die oberen RAM-Adressen belegen. Die entsprechenden Werte für START und die jeweiligen Anfangsadressen des Programms finden Sie in Tabelle 3. Die Adresse 'INI' ergibt sich immer aus Start * 256 + 76. Für den nicht erweiterten VC-20 also: 29 * 256 + 76 = 7500.

Mit diesem Maschinenprogramm läßt sich die Centronics-Schnittstelle wie der eingebaute Commodore-Druckeranschluß bedienen. Die Befehlsfolgen in Tabelle 4 zeigen entsprechende Beispiele.

```
SYS (INI)
OPEN 1 , 5
PRINT#1,"DRUCKEN MIT CENTRONICS"
CLOSE 1
```

oder

```
SYS (INI)
OPEN 1 , 5
CMD 1
PRINT "DIESES IST DER DRUCKER"
PRINT#1
PRINT "DIESES IST DER BILDSCHIRM"
CLOSE 1
```

Tabelle 4. Beispielprogramme, die die Schnittstelle ansprechen

Retten auf Datasette

Nachdem Sie das Maschinenprogramm nun mit viel Mühe in den Speicher 'gepokt' haben, hier noch ein Tip für das Retten auf die Datasette. Dazu verwenden Sie bitte die Befehlssequenz in Tabelle 5. Die Befehlsfolge fügt zwei Nullbytes vor dem Maschinenprogramm ein und speichert es auf der Datasette ab. Die beiden Nullbytes sind notwendig, damit der Zeilenlinker das Programm beim Laden nicht 'umgräbt'.

Geladen wird das Programm mit:

```
LOAD "C",1,1
SYS (INI)
NEW
```

Um das Ganze mit dem c't-Sprachsynthesizer auszuprobieren, könnten Sie einmal den folgenden Text ausgeben lassen:

```
CÖwwjFMCjUHÖ
CCCCCCCjÜAMjkuMi
YÜCiÜjCYXHÜÜBCC
```

dabei bedeuten:

Ö = Commodoretast + '—'
Ü = Shift + '—'

Damit die Sprachplatine diesen Text verarbeiten kann, muß der Rechner in den Kleinschreibungsmode geschaltet werden.

```
POKE 43,254
POKE 44,START — 1
POKE 45,128
POKE 46,START
POKE (256*START)—2 , 0
POKE (256*START)—1 , 0
SAVE "CENTRONICS"
```

Tabelle 5. Befehlsfolge, um das Programm abzuspeichern.

nascom-C

CP/M ist ein eingetragenes Warenzeichen der Firma Digital Research

Jumbo-Komfort für Ihren Tischrechner

bietet Ihnen das neue CP/M 3.0 Betriebssystem!

NASCOM-C ist maßgeschneidert für CP/M3

Kompatibilität und Systemkomfort.
NASCOM-C ist ein neues deutsches System zur bestmöglichen und komfortabelsten Ausnutzung der modernsten CP/M-Software-Produkte.

Ergonomie für jedermann erschwinglich.
Ein System, das einiges auf dem Kasten hat und trotzdem viel preiswerter ist als so mancher Homecomputer, vor allem voll ausgebaut. Es entspricht den deutschen Vorstellungen von Benutzerfreundlichkeit, Service und Ergonomie. Ob Sekretärin, Handwerker, Student, Tüftler, Techniker oder Manager: Jeder findet in NASCOM-C einen zuverlässigen Freund, der allzeit bereit nervtötende Routinearbeit übernimmt und seinen menschlichen Partner versteht anstatt ihn zu veräppeln.

Ein System, das mitdenkt.
NASCOM-C versteht Sie, auch wenn Sie bisher nichts über Computer wissen, hilft Ihnen aber auch, Computer zu verstehen, hat keine Geheimnisse. Das geht so weit, daß Sie ihn auch selbst zusammenbauen und so von Grund auf kennenlernen können. Wer seinem NASCOM-C ein bißchen Zeit widmet, kann so viel Geld sparen und noch mehr lernen.

Gute Software braucht bessere Hardware!
NASCOM-C bedient sich des neuen CP/M-Plus-Betriebssystems, das bisher auf Mikros nicht bekannten Komfort mit der Kompatibilität zu seinen Vorläufern verbindet, für welche das größte Angebot an hochwertiger Software existiert. Aber was nützt die schönste Software, wenn die vorhandene Hardware deren Möglichkeiten nicht nutzen kann? Faule Kompromisse auf der Hardwareseite sind bei CP/M+ nämlich schwerlich möglich. Bessere Betriebssysteme brauchen nun mal modernere Computer. NASCOM-C ist ein für CP/M+ maßgeschneiderter Rechner, der allen alle Möglichkeiten von CP/M+ bietet, ohne Nerven, Geldbeutel oder Spezialkenntnisse zu überfordern.

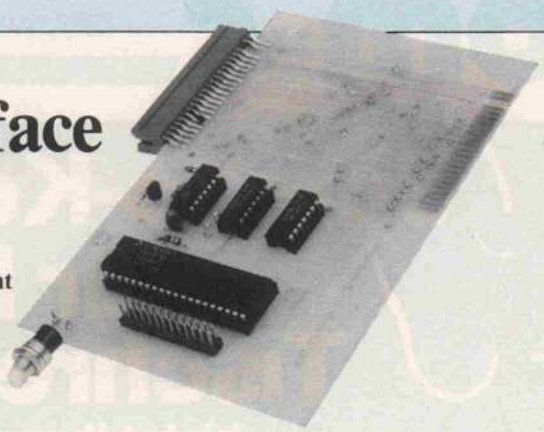
Wer bietet mehr auf einer Karte?
- Z80 A/B Zentraleinheit mit 4/6 MHz Takt
- Speicherverwaltungseinheit (MMU) und DMA
- 128 Kilobyte Arbeitsspeicher mit Paritätsprüfung, erweiterbar auf 1 Megabyte
- Floppy-Disk Controller für alle 5¼- und 8-Zoll Laufwerke

- Festplattenschnittstelle vorhanden
- Ein DEC VT52, -HEATH H19 -und ANSI-Aufwärtskompatibles Terminal mit Grafik in 8 Farben und ladbarem Zeichengenerator
- Zwei V.24 und eine Centronics-Schnittstelle zum Anschluß von Druckern, Plottern, Modems und Hostrechnern
- Über RS-422 Schnittstellen zum Netzwerk erweiterbar
- Der 77-polige NASBUS macht NASCOM-C kompatibel zu vielen Erweiterungskarten (wie Farbgraphik mit 792*256 Punkten und 256 KB Speichererweiterungen
- Farbgraphik kompatibel zu den Normen Tektronix 40XX, Plot-10 und GKS, umfangreicht Objektcode-Bibliotheken
- Jede wichtige Programmiersprache verfügbar
- Branchenlösungen, Text- und Datenbanksysteme in kaum überschaubarer Vielfalt
- Von der Leerplatine mit Dokumentation und Firmware bis zum Fertiggerät lieferbar
- Der Grundbausatz einschließlich Dokumentation und Firmware kostet unter DM 1.000,–, als Leerplatine sogar nur DM 298,–!!

Wer lieber gleich den richtigen Computer kaufen will oder den Frust mit seinem jetzigen satt hat, bekommt für 2 DM sein NASCOM-C INFO-Paket direkt von

LAMPSON & ZERBE GmbH
Mikrocomputervertrieb
Odenwaldstraße 21–23
6087 Büttelborn · Tel. 0 61 52/5 67 30

Centronics-Interface für ZX81



Wer die Ausdrucke des Sinclair-Computers nicht mehr auf dem vornehmen Silberpapier, sondern schlicht schwarz auf weiß, dafür aber in DIN-A4-Format haben möchte, stößt auf ungeahnte Schwierigkeiten. Vor den Anschluß eines handelsüblichen Druckers hat Sinclair einige Hindernisse gestellt. Soll also ein Drucker mit Centronics-Schnittstelle vom Sinclair gesteuert werden, muß ein Interface Vermittlerdienste leisten.

A. Stiller

Der ZX81 bereitet alle nur denkbaren Schwierigkeiten bei der Realisierung dieser Schnittstelle. Die I/O-Ports können nicht ohne weiteres genutzt werden, da die vom Rechner genutzten Portadressen nicht sauber ausdekodiert sind. Weil der ZX81 den ASCII-Code nicht kennt, muß die Treibersoftware den Sinclair-Code übersetzen. Da die ZX-Software zur Druckersteuerung unerreichbar im ROM liegt, können Befehle wie 'LPRINT' nicht zur Steuerung der Centronics-Schnittstelle verwendet werden. Es ergeben sich also einige Probleme, die es zu lösen gilt.

Vier ICs führen zum Drucker

Die eigentliche Schnittstelle zwischen dem ZX81 und dem Drucker bildet ein PIO-Baustein (Parallel Input/Output). Dieser stellt drei 8-Bit-Ports bereit, wovon die Schnittstelle zwei belegt. Soll an die PIO kein Drucker angeschlossen werden, können die Ports natürlich frei verwendet werden. Aber auch, wenn ein Drucker betrieben wird, kann man über einen Port frei verfügen.

Die drei anderen ICs dekodieren die Portadresse der PIO. Aus dem eingangs erwähnten Problem ergibt sich, daß die PIO nicht als normaler I/O-Baustein angesprochen werden kann. Als Ausweg bietet sich an, den Port 'Memory-Mapped' zu betreiben. Das bedeutet, daß die PIO wie normaler Speicher angesprochen wird. Dies bringt zusätzlich den Vorteil, daß die Befehle 'PEEK' und 'POKE' zur Steuerung der PIO verwendet werden können. Als Nachteil ergibt sich, daß die Adresse, auf der die PIO liegt, nun nicht mehr als Speicherplatz benutzt werden kann. Daß der Port Memory-Mapped arbeitet, erreicht man dadurch, daß der Pegel der Leitung $\overline{MREQ}$ in die Adreßcodierung mit eingeht. Ist der Pegel dieser Leitung 'Low', zeigt die CPU an, daß sie einen Schreib-/Lesezugriff auf den Speicher vornehmen will.

Die Dekodierung der Portadresse geschieht also nach folgendem Schema:

$\overline{MREQ}$ A15 ... A7 ... A0

0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 X X

Das entspricht den Adressen 29C0h...29C3h (10688d...10691d). Die Adressen A0 und A1 werden von der PIO selbst dekodiert, so daß sie in diese Auswertung nicht mit eingehen.

Um zu verhindern, daß auf dieser Adresse das ROM des ZX als 'Spiegelbild' erscheint (bedingt durch eine unvollständige Adreßdekodierung im Computer), muß immer, wenn die Portadresse selektiert ist, das ROM des Sinclair abgeschaltet werden. Das ermöglicht die Leitung $\overline{ROMCS}$ des Computers. Liegt hier ein Signal mit dem logischen Pegel 1 an, ist das ROM abgeschaltet. Für diese Abschaltung ist aber etwas Leistung nötig, da der Sinclair intern eine merkwürdige Schaltung hat: Der Ausgang des Steuerbausteins ULA liegt über einen 680-Ohm-Widerstand an dem $\overline{CS}$-Eingang des ROMs. Selektiert die ULA das ROM durch ein Signal mit dem logischen Pegel 'Low' und soll das ROM gleichzeitig mit einem High-Signal an der Kontaktleiste (Pin $\overline{ROMCS}$) 'deselektiert' werden, so fließt über den Widerstand ein Strom von mindestens 3,5 mA. Diesen Strom bringt bei der Interface-Karte der Transistor T1 auf.

Nicht nur drucken

In diesem Artikel geht es zwar um die Frage, wie mit der PIO-Karte eine Centronics-Schnittstelle realisiert werden kann; es ergeben sich aber daneben noch eine ganze Reihe nützlicher Anwendungen für die Platine.

Der Taster 'RESET' bewirkt einen Reset des Computers. Damit befindet sich der Rechner in dem Zustand, in dem er nach dem Einschalten der Stromversorgung ist.

An die Kontaktleiste an der Platine können alle üblichen Erweiterungen des ZX81 gesteckt werden, da die Belegung dieser Leiste der des Sinclairs entspricht.

Um die Ports der PIO für freie Anwendungen zu nutzen, muß man einiges über die Programmierung der PIO wissen. Hier also ein Kurzlehrgang in Sachen PIO:

Die PIO 8255 verfügt über drei 8-Bit-Ports, die unabhängig voneinander als Ein- oder Ausgabeport definiert werden können. Dazu wird in das Steuerregister der PIO ein Steuerwort geschrieben, das die gewünschte Portdefinition ergibt (siehe Tabelle 1). Ist ein Drucker angeschlossen, sollte man darauf achten, daß Port A als Ausgang, Port C, Bit 0...3 als Eingang und Port C, Bit 4...7 als Ausgang definiert bleiben.

Steuerwort (dez.)	Port A	Port B	Port C Bit 0...3	Port C Bit 4...7
128	out	out	out	out
129	out	out	in	out
130	out	in	out	out
131	out	in	in	out
136	out	out	out	in
137	out	out	in	in
138	out	in	out	in
139	out	in	in	in
144	in	out	out	out
145	in	out	in	out
146	in	in	out	out
147	in	in	in	out
152	in	out	out	in
153	in	out	in	in
154	in	in	out	in
155	in	in	in	in

Tabelle 1. Steuerworte der PIO

	A	B
1	PA4	PA3
2	PA5	PA2
3	PA6	PA1
4	PA7	PA0
5	GND	PC7
6	+5V	PC6
7	PB3	PC5
8	PB4	PC4
9	PB5	PC0
10	PB6	PC1
11	PB7	PC2
12	PB2	PC3
13	PB1	PB0

Tabelle 2. Belegung der Port-Steckerleiste X1

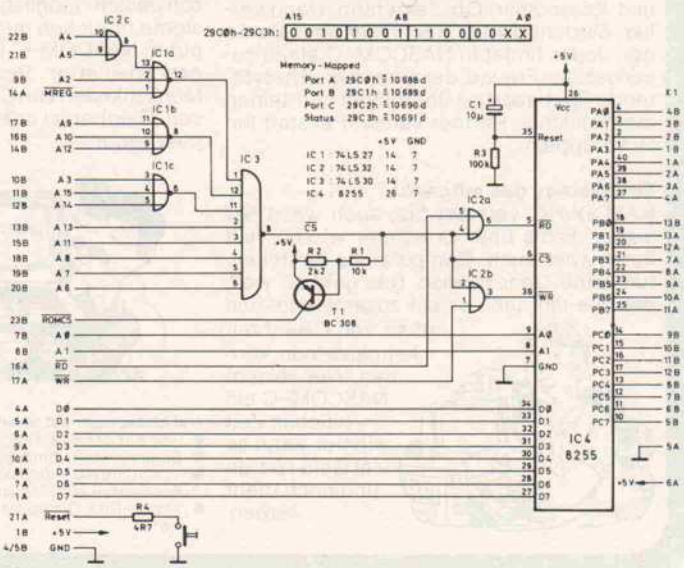


Bild 1. Die Werte der Bauteile können dem Schaltbild entnommen werden.

DR. AUMANN

CP/M 86 f. PC/XT			205 DM
TEAC-Floppy-Disk-Laufwerke: Ganz neu im Programm, 3″-Laufwerke anschließbar wie 5″-Laufwerke, vollständig kompatibel.			
3″-Laufwerke	FD 30 A	40 Spuren einseitig	550 DM
5″-Laufwerke	FD 55 A		640 DM
(alle in SLIMLINE-	FD 55 B		750 DM
Ausführung)	FD 55 E		740 DM
	FD 55 F		880 DM
	FD 55 G	1,66 Mio. Byte	1040 DM
Genie 16, IBM-kompatibel 8086, 256 K, 2 x 360 KB Disk, HyRes-Graphik, flache freibewegliche Tastatur			5900 DM

Markendisketten ab DM 5,47 (10er-Preis) mit Verstärkungsring. Wir liefern auch Steckverbinder, Bauteile, Drucker zu interessanten Preisen. Alle Preise incl. MwSt., Versand ab Straßenhaus.
Fordern Sie unsere Infos an!!!

Dr. Aumann GmbH Computersysteme, Schulstraße 12, 5451 Straßenhaus
Tel. 0 26 34/40 81, Bürozeiten: montags — freitags 9—12/14—17 Uhr

Ihr Computer versteht auch Spaß . . .
. . . Testen Sie es!!!

mit SUPERSPIELEN für

ZX SPECTRUM

Mined Out	48 k (C) DM 29,—	Space Intruders	(C) DM 29,—
The Hobbit	48 k (C) DM 78,—	Masterfile	48 k (C) DM 48,—
Penetrator	48 k (C) DM 37,—	Display	16/48 k (C) DM 28,—
Terror Daktil	48 k (C) DM 37,—	Matcalc	16/48 k (C) DM 39,—
Time Gate	48 k (C) DM 37,—	Monitor	16/48 k (C) DM 30,—
Jet Pac	16 k (C) DM 31,—	Edit./Ass.	16/48 k (C) DM 35,—
BASIC COMPILER von Softek			48 k (C) DM 69,—

Weitere Spiele für ZX 81, VC 20, VC 64, APPLE, IBM PC

HARDWARE

ZX Spectrum mit 48 k und deutschem Handbuch!!!	DM 525,—
Centronics Interface für ZX Spectrum	DM 170,—
Zusatztastatur (Spectrum wird eingebaut)	DM 190,—
COMPETITION-PRO für Atari 400/800/VC 20/VC 64	DM 69,—
SUPER-JOYSTICK mit Interface für ZX SPECTRUM	DM 129,—
ERGOTILT! der ergonomische Monitor-Untersatz macht Ihren Monitor dreh- u. kippbar!!	DM 98,—

Preise inkl. 14 % MwSt. Versand per Nachnahme

STEDE Spezialversand **Postfach 12 66, 3542 Willingen**

CEPAC-80

CMOS-Computer (wie in c't 1/84 beschrieben) mit Z80-kompatibler CPU, 3 Timern und 48 I/O-Leitungen.

Fertiggerät auf Europaplatinen mit NSC 810 und 27C64 . **298,— DM**

Platine 100 x 86 mm	**49,— DM**	NSC 800	**48,— DM**
Platine 100 x 160 mm	**69,— DM**	NSC 810	**68,— DM**
Bauteilesatz		8155	**15,80 DM**
(Mit CPU [1 MHz] ohne Platine,		6116 LP-3	**15,80 DM**
Speicher, Riot)	**99,— DM**	2764	**19,80 DM**
Fertig aufgebaut,		27C16	**42,50 DM**
Aufpreis	**40,— DM**	NE 590	**11,80 DM**

Alle Preise inkl. MwSt. Info gratis.

CONITEC **Christian Lotter KG**

Postfach 11 06 22, Schuchardstraße 4
6100 Darmstadt, Telefon (0 61 51) 2 60 13

Wir haben die Lösung

Floppy-Laufwerk
Halbspur und Disketten wenden 40 Track
jetzt nur noch . . . . . **699,— DM**

Floppy-Controller
vorbereitet für bis zu 2 x 80 Track DS-Betrieb
jetzt für . . . . . **199,— DM**
APPLE®-kompatibel-Rechner

Profi-Version
großes Netzteil, großes Gehäuse, Preh AK 88 Tastat., Groß- und Kleinschreibung, deutsche Umlaute, jetzt nur . . . . **1499,— DM**

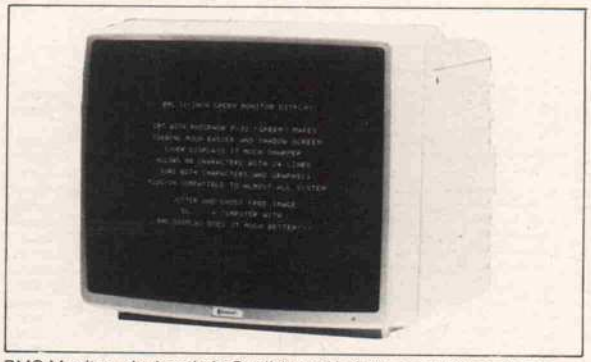
Teac-Floppy FDS5 A 40 Track . . . . . **639,— DM**
Disketten für APPLE, 10 St. . . . . **62,— DM**

Liste kostenlos!

UEDING ELECTRONICS
5750 Menden 1 · Holtewiese 2 · Telefon (0 23 73) 6 31 59

BMC Neue Produkte ab 1984:

BM 8181	Farbmonitor RGB 640 x 240 Pkt. für IBM	DM 1584,60
BM 12 EN	hohe Bandbreite v. 20 MHz, grün, entspiegelt	DM 498,—
BM 12 EY	Bandbreite > 18 MHz, bernstein	DM 438,—
BM 12 A	Bandbreite > 15 MHz, grün	DM 298,—

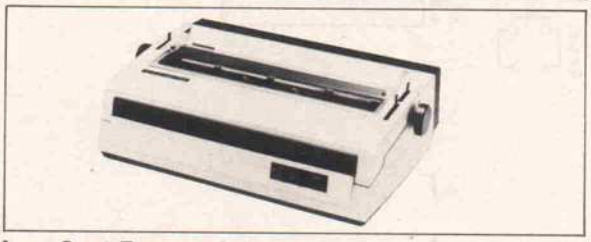


BMC-Monitor, einzigartig in Qualität und Leistung, mit Filterscheibe, > 18 MHz, grüner Röhre, im formschönen Datenmonitor-Gehäuse

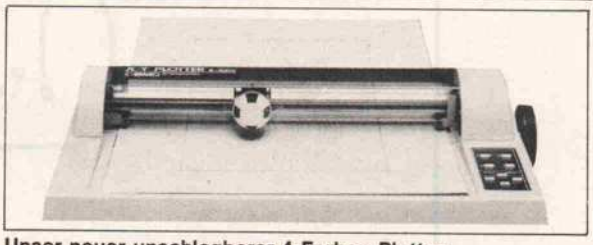
BM 12 ES = 398,— DM inkl. MwSt.
(349,12 DM netto) Dazu passend ERGOTILT 89,— DM inkl. MwSt.

Typenrad-Drucker mit 40 Z/s
Friktion + Traktor,
380 mm Papierbreite,
Adaptionsmöglichkeit
für Sheet-Feeder,
Standard-8-bit-Schnittstelle
Optional RS 232 lieferbar

SD 1152:
4390,— DM inkl. MwSt.
(netto 3850,88 DM)



Low-Cost-Typenrad-Drucker
16 CpS, Schreibbreite 335 mm, 96-Z-Typenrad, Friktionsführung, mit Einzelblatteinzug, 8-bit-Schnittstelle 2-KB-Puffer
TD 16 = 2490,— DM inkl. MwSt.
(netto) 2184,21 DM)

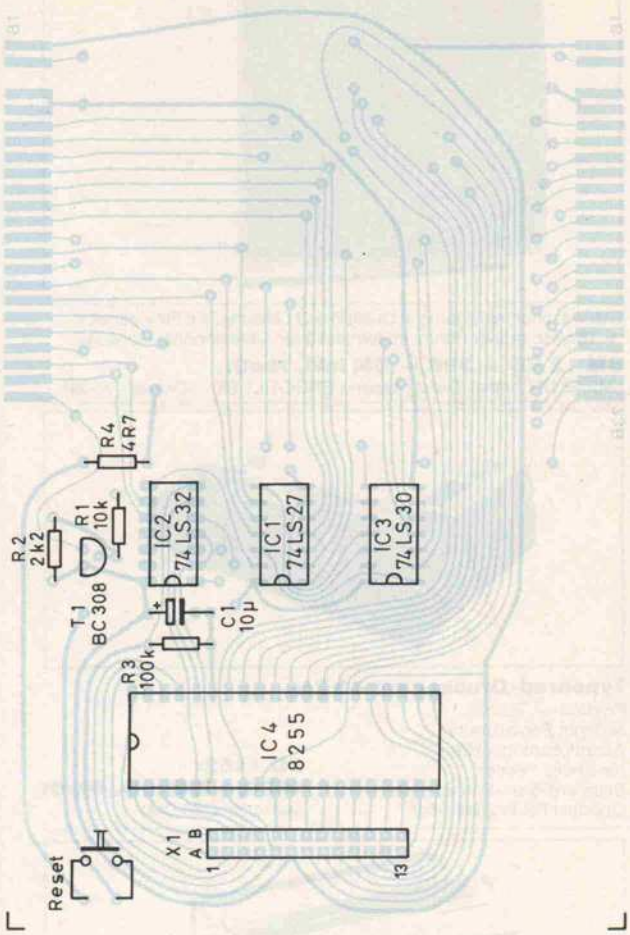


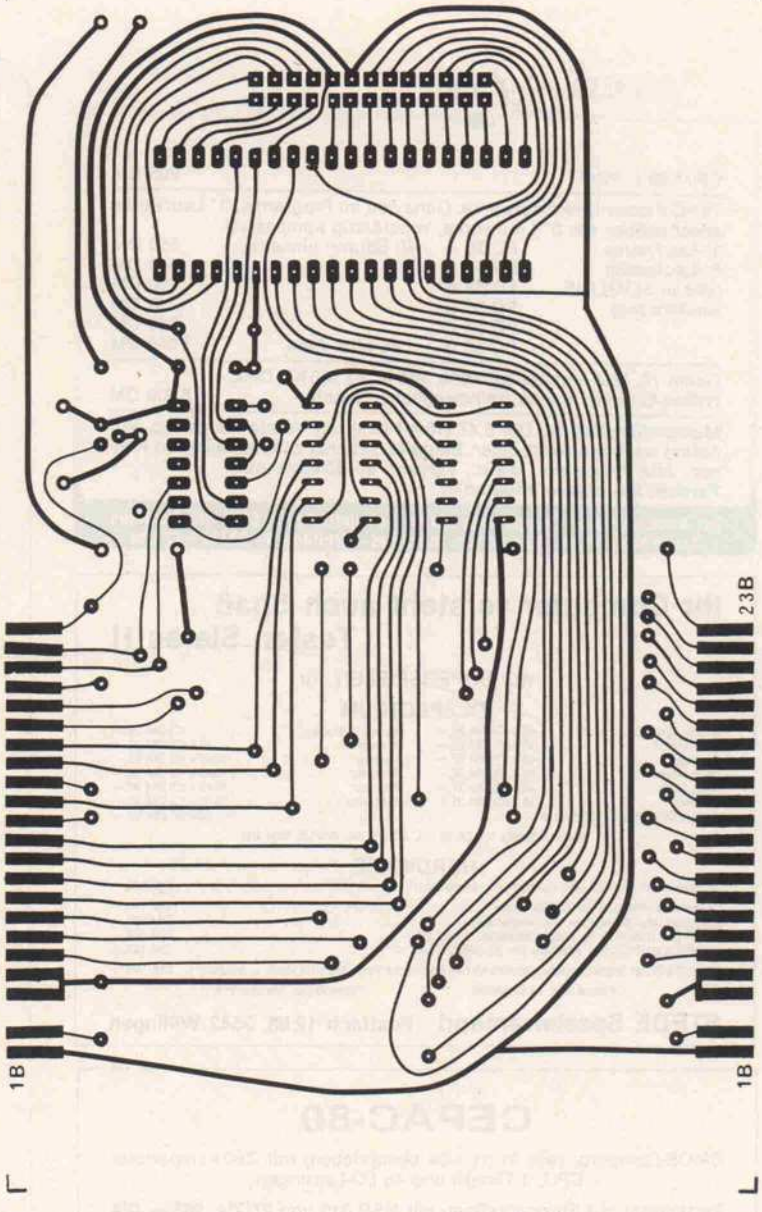
Unser neuer unschlagbarer 4-Farben-Plotter
DIN A3-Format, 0,1 mm-Genauigkeit, Schreibgeschwindigkeit 100 mm/s, ASCII-Zeichensatz u. Kreisfunktion! Optional Graph-ROM!
MP 1003 2690,— DM inkl. MwSt.
(netto 2359,64 DM)

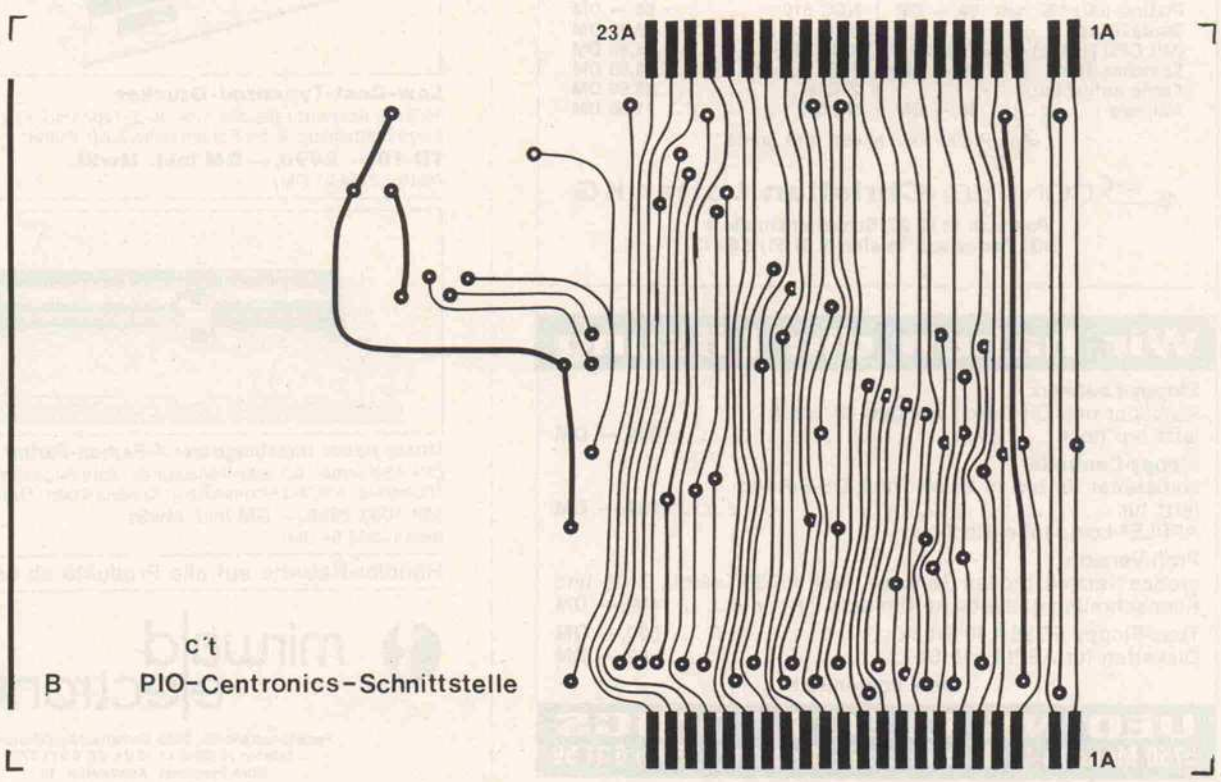
Händler-Rabatte auf alle Produkte ab dem 1. Stück!

mirwald electronic **BMC**

Fasanenstraße 8b, 8025 Unterhaching/München,
Telefon (0 89) 6 11 12 24, FS 5 213 476
Büro Frankfurt: Adalbertstr. 15
Telefon (06 11) 70 35 38







c't
B PIO-Centronics-Schnittstelle

PEANUT-PROFI DS-64
Doppelprozessorsystem
mit 6502 und Z80A

12kB ROM, 64kB RAM
für CP/M umd Applesoftware
Tastatur mit eigenem Prozessor
10 Funktionstasten
PAL und RGB-Ausgang

Komplett-System mit 2 integr.
Slimline-Floppies DM 2948,00
mit 1 Laufwerk DM 2598,00

IEC-Bus Interface
(auch für apple) DM 698,00

Sämtliche IF-Karten für Apple
z.B. 80-Zeichen DM 249,00
16 Kanal A/D + D/A-Wandler
DM 448,00

Sortierprogramm für VISICALC
SORTITEC DM 410,40

Original CP/M Software
zu Sonderpreisen

Info gegen 1,30 Porto
Händlerkonditionen auf Anfrage

IMETEC EDV
6200 Wiesbaden
Wasserrolle 2
✆ 06121/24622

FUTTER
für den C64

Ihr Computer ist ohne Programme wie ein Auto ohne Benzin.

Gute und preiswerte Programme für Ihren C64 bieten wir mit dem SYNTAX-Programm-Kassetten-Magazin.

Jeden Monat erscheint eine Kassette mit 6 neuen, vielseitigen Programmen für Ihren C64.

SYNTAX-Programme auf Kassetten und Disketten sind auch für die Commodore CBM und VC 20 erhältlich. Nutzen Sie Ihr Gerät verstärkt durch neue Ideen.

Fordern Sie gleich heute noch unter Angabe Ihres Gerätetyps kostenlose Informationen von

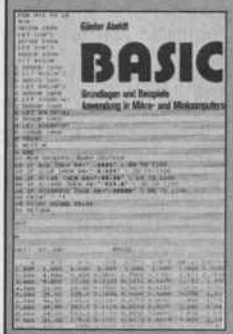
SYNTAX
Soft- u. Hardware GmbH
P.B. 18 68, 7550 Rastatt
Telefon (0 72 22) 7 23 15

TOPP aktuell

Best.-Nr. 354
D. Böhm
Ich möchte einen Computer
DM 10,80

Best.-Nr. 355
D. Böhm
Computergesteuerte Meßtechnik
DM 25,80

Best.-Nr. 428
J. Kwiatkowski
FORTRAN in 8 Lektionen für Anfänger
DM 29,80



Best.-Nr. 455
G. Abeldt
BASIC – Grundlagen und Beispiele
DM 9,–

Best.-Nr. 401
Rowley
Atari BASIC
DM 10,80

Best.-Nr. 496
Libes/Wahl
Personal Computer Handbuch
DM 19,80

Best.-Nr. 475
M. Reinhard
Mikroprozessor SCMP II
DM 19,80

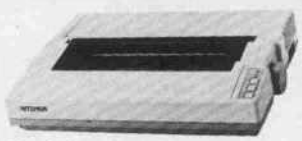
Best.-Nr. 420
L. Schüssler
Fernsteuern – Sonder- u. Zusatzfunktionen
DM 19,80

Best.-Nr. 424
K. Diedrich
Synthesizer
DM 14,80

Diese informative Buchreihe von TOPP wird ständig durch weitere Titel ergänzt, die zum Verständnis unserer Tage beitragen.
Prospekte über die Gebiete Elektronik und Amateurfunk stehen kostenlos zur Verfügung. – Bitte anfordern.

frech-verlag
Turbinenstr. 7, 7000 Stuttgart 31
Telefon (0711) *83 20 61, Telex 7 252 156 fr d

RITEMAN



Matrixdrucker 120 cps
parallel-Centronics-Schnittstelle, Bidirektional-Druck, graphikfähig 960 dpl . . . . . **DM 1150,–**

Drucker von C. ITOH.

8510 A Matrixdrucker 120 cps
parallel mit dt. Handbuch für Endlos und Einzelblatt . . . . . **DM 1499,–**

8510 S Matrixdrucker 180 cps
freiladbarer Zeichensatz und 5 zusätzliche Zeichensätze **DM 1999,–**

Super-Graphik-Interface
für alle Itoh-Geräte in parallel, 32seitiges Interface-Handbuch für Apple-Computer u. kompatible **DM 199,–**

BASE-48
„Apple-kompatibler" Computer 48 K mit Groß- und Kleinschrift, original Gehäuse – Handbuch . . **DM 999,–**

Elektronik-Vertrieb KÖLLER
COMPUTER UND ZUBEHÖR
Lothe, Niesetalstraße 4
4938 Schieder-Schwalenberg
Telefon 0 52 33 / 75 50

Preise inkl. MwSt.
Versand per NN plus Porto und Verpackung

ZX 81 und ZX Spectrum
Zubehör von Logitek

Druckerinterface für ZX-Spectrum
Anschlußfertig für fast alle erhältliche Drucker wie EPSON, STAR, CP80 usw. LPRINT, LLIST und COPY sind sofort verfügbar. Viele zusätzliche Druckfunktionen sind in BASIC ansprechbar, wie wählbare Schriftstärke und Schriftbreite, direkte ASCII-Ausgabe über LPRINT, Erweiterung des Spectrumzeichensatzes mit deutschen Zeichen, in Verbindung mit der 80 K Speichererweiterung ist während des Druckens zugleich ein Weiterarbeiten am Rechner möglich.
Komplett mit Kabel **DM 275,—**

LOGITEK Spectrumgehäuse
Dieses formschöne schwarz eloxierte Aluminiumgehäuse nimmt den Spectrum mit Netzteil und Busplatine für 5 Erweiterungen auf. Mit EIN/AUS Schalter.
Gehäuse komplett mit Bus **DM 238,—**
Busplatine für 5 Karten **DM 118,—**

Speichererweiterung von 16 K **auf 80 K** zum Einstecken in den Spectrum. **DM 228,—**
(Bitte bei Bestellung Issue 2 oder 3 angeben!)

32 Bit Portmodul
Über die 32 Leitungen lassen sich elektronische Steuer-, Regel- und Meßschaltungen anschließen, die digitale Ein- und Ausgänge haben.
Für ZX-Spectrum und ZX-81 **DM 138,—**

Spectrumstecker **DM 14,—**
Gegenstück dazu **DM 7,—**

64 K RAM Modul für ZX-81
schwarz elox. Alugehäuse, flach an den ZX-81 ansteckbar, Port durchgeführt. **DM 238,—**

ZX-81 Stecker **DM 12,—**
Gegenstück dazu **DM 6,—**

Räumliche Grafik für ZX Spectrum
Mit diesem Programm lassen sich mathematische Funktionen räumlich darstellen und anschaulich erklären. **DM 38,—**

Weitere Programmkassetten auf Anfrage.
Datenleerkassette C30 **DM 2,—**

Deutsche Beschreibungen werden mitgeliefert. Preise incl. MwSt. Versand per NN zzgl. 6,50 DM. Porto und Verpackung ab Lager Berlin.

LOGITEK
Höft und Lesser GbR
Pankstraße 49, 1000 Berlin 65
Telefon (030) 461 64 92

Centronics-Interface

Die Software für den ZX81

Port B wird für die Druckersteuerung nicht gebraucht, steht also dem Benutzer für andere Zwecke zur Verfügung. Es kommen also die Steuerwörter 129 und 131 in Betracht, wobei wir für das vorliegende Programm 129 (81h in Zeile 9) ausgewählt haben. Bit 0 von Port C soll als BUSY und Bit 7 als DATA STROBE angeschlossen werden.

Zum Betreiben der Centronics-Schnittstelle können die Druckerbefehle des ZX81 leider nicht verwendet werden. Diese arbeiten im Grafik-Mode und geben genau wie bei der Bildschirmausgabe Pixel für Pixel aus. Dabei ist die Druckzeile auf 32 Zeichen beschränkt.

Das Programm PRCENT erzeugt aus einem ZX81-String einen ASCII-String. Alle ASCII-Zeichen können ohne CHR$-Befehl aus dem ZX81-Zeichensatz generiert werden (Tabelle 3). Als Besonderheit enthält der ZX81-Zeichensatz nur das Pfund-Symbol — es handelt sich ja schließlich um ein britisches Gerät — und die Blockgrafik-Zeichen. Diesen Symbolen sind in der Umformtabelle andere ASCII-Zeichen zugeordnet. Die Inversbuchstaben des ZX werden als Kleinbuchstaben interpretiert. Der Rest wird auf Sonder- oder Steuerzeichen abgebildet. Für den Drucker sind dabei das inverse $ als 'Carriage Return' und das Blockgrafikzeichen über grafic shift F als 'Linefeed' von Bedeutung. Diese Zeichen müssen also als Abschluß einer Printzeile in dem Ausgabestring auftauchen. Für die Ansteuerung des c't-Sprachsynthesizers sind diese Steuerzeichen natürlich nicht nötig.

```
 2 REM =7B00
 6 REM E7
 7 REM CD550F
 8 REM CDF813
 9 REM 3E81
10 REM 32C329
11 REM 3D
12 REM 32C229
13 REM 1A
14 REM CD1F7B
15 REM CD787B
16 REM 13
17 REM 0B
18 REM 78
19 REM B1
20 REM 20F3
21 REM CF
22 REM FF
27 REM FEC0
28 REM 2826
29 REM CBB7
30 REM FE80
31 REM 380C
```

Bild 2. ZX81 druckt den Anfang des Druckprogramms

Die Stringeingabe erfolgt mit

```
PRINT CHR$ USR
31488 = ".....",
```

wobei der String auch in einer Variablen enthalten sein kann, zum Beispiel:

```
PRINT CHR$ USR
31488 = X$
```

Für die Eingabe des Maschinenprogramms empfiehlt es sich, den Mikroassembler aus dem SuperTape-Artikel im gleichen Heft zu verwenden. In Spalte drei des Assembler-Listings befindet sich dann die Zeilennummer der REM-Statements, und Spalte zwei enthält den Hexcode. Als Startadresse ist 7B00 für die Ausgaberoutine PRCENT gewählt. Vor der Eingabe und vor jedem Laden von der Kassette sollte dieser Bereich mit

```
POKE 16389,123
NEW
```

geschützt werden.

Das Programm hat vier Aufgaben zu erfüllen: String-Startadresse und -länge zu bestimmen, den String abzuarbeiten, das jeweilige Zeichen in ASCII umzuwandeln und schließlich an die Schnittstelle auszugeben. Die ASCII-Zeichen werden bei Buchstaben und Steuerzeichen berechnet; bei Sonderzeichen dient eine Tabelle zur Umwandlung.

Die Druckerausgabe folgt im wesentlichen dem Flußdiagramm in dem Artikel über parallele Schnittstellen in c't 1/84. Auf die PAPER OUT — Abfrage wurde dabei verzichtet, dafür aber eine BREAK-Abfrage mit einbezogen.

	0	1	2	3		8	9	A	B
0	SP	(	4	K	0	NUL	DLE	4 `	K k
1	£	)	5	L	1	SOH	DC1	5 a	L l
2	½	>	6	M	2	STX	DC2	6 ö	M m
3	ü	<	7	N	3	ETX	DC3	7 ü	N n
4	^	=	8	O	4	EOT	DC4	8 ß	O o
5	_	+	9	P	5	ENQ	NAK	9 DEL	P p
6	!	-	A	Q	6	ACK	SYN	A a	Q q
7	#	*	B	R	7	BEL	ETB	B b	R r
8	$	/	C	S	8	BS	CAN	C c	S s
9	'	;	D	T	9	HT	EM	D d	T t
A	§	,	E	U	A	LF	SUB	E e	U u
B	"	.	F	V	B	VT	ESC	F f	V v
C	£ &	0	G	W	C	FF	FS	G g	W w
D	$	1	H	X	D	CR	GS	H h	X x
E	:	2	I	Y	E	SO	RS	I i	Y y
F	?	3	J	Z	F	SI	US	J j	Z z

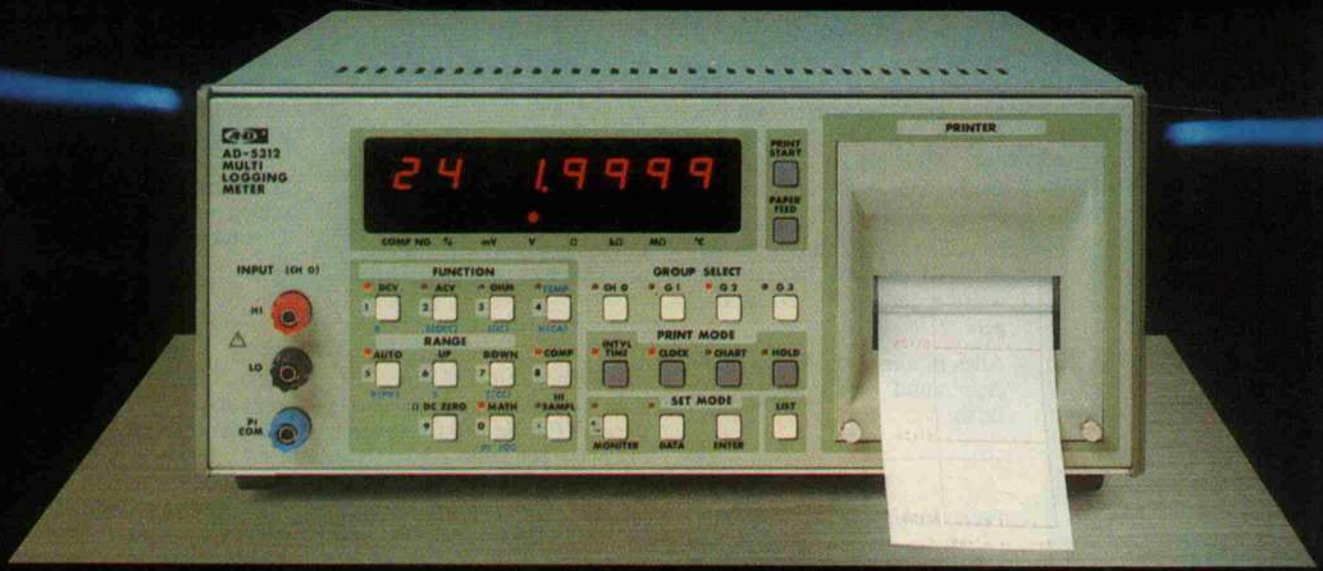
Tabelle 3. Umwandlung ZX-Code in ASCII (deutscher Zeichensatz)

```
                    0001  ;
          (7B00)    0002        ORG     7B00H
                    0003  ;-------------------------------------------------
                    0004  ;-----  *****   PRCENT  *****  -------------------
                    0005  ;-------------------------------------------------
7B00  E7            0006  PRCENT  RST     20H             ;ÜBERSPRINGT "="
7B01  CD550F        0007        CALL    0F55H           ;BEST. DE=STRINGADR,
7B04  CDF813        0008        CALL    13F8H           ;       BC=LÄNGE
7B07  3E81          0009        LD      A,81H           ;SETZT PORTKONFIG
7B09  32C329        0010        LD      (29C3H),A       ;       "
7B0C  3D            0011        DEC     A               ;SETZT STROBE-FLAG
7B0D  32C229        0012        LD      (29C2H),A       ;       "
7B10  1A            0013  LOOP    LD      A,(DE)          ;CHAR. VOM STRING
7B11  CD1F7B        0014        CALL    ZXASCII         ;ASCII-WANDLUNG
7B14  CD787B        0015        CALL    LPRCHAR         ;PORTAUSGABE
7B17  13            0016        INC     DE              ;INKR. STRINGADR
7B18  0B            0017        DEC     BC              ;DEKR. SCHLEIFENZÄHLER
7B19  78            0018        LD      A,B
7B1A  B1            0019        OR      C
7B1B  20F3          0020        JR NZ,  LOOP            ;SPRUNG, FALLS <>0
7B1D  CF            0021        RST     08H             ;ENDE DES AUFRUFS MIT
7B1E  FF            0022        DEFB    0FFH            ;ERR-CODE = 0
                    0023  ;
                    0024  ;-------------------------------------------------
                    0025  ;-----  *****   ZXASCII *****  -------------------
                    0026  ;-------------------------------------------------
7B1F  FEC0          0027  ZXASCII CP      0C0H            ;TEST AUF
7B21  2826          0028        JR Z,   ANF             ;ANFÜHRUNGSZEICHEN
7B23  CBB7          0029        RES     6,A
7B25  FE80          0030        CP      80H             ;TEST AUF NORMAL ODER
7B27  380C          0031        JR C,   NORMAL          ;INVERS
7B29  D645          0032  INVERS  SUB     45H
7B2B  FE61          0033        CP      61H             ;RETURN FALLS KLEIN-
7B2D  D0            0034        RET NC                  ;BUCHSTABE
7B2E  D63B          0035        SUB     3BH
7B30  FE20          0036        CP      20H             ;RETURN FALLS
7B32  D8            0037        RET C                   ;STEUERZEICHEN
7B33  1807          0038        JR      SONDER          ;SONST: SONDERZEICHEN
7B35  C61B          0039  NORMAL  ADD     1BH
7B37  FE41          0040        CP      41H             ;RETURN FALLS
7B39  D0            0041        RET NC                  ;BUCHSTABE
7B3A  D61B          0042        SUB     1BH
7B3C  E5            0043  SONDER  PUSH    HL              ;SONDERZEICHEN
7B3D  C5            0044        PUSH    BC
7B3E  214C7B        0045        LD      HL,TAB
7B41  0600          0046        LD      B,00H           ;BEST. DES ASCII-
7B43  4F            0047        LD      C,A             ;ZEICHENS AUS DER
7B44  09            0048        ADD     HL,BC           ;TABELLE
7B45  7E            0049        LD      A,(HL)
7B46  C1            0050        POP     BC
7B47  E1            0051        POP     HL
7B48  C9            0052        RET
                    0053  ;-------------------------------------------------
7B49  3E22          0054  ANF     LD      A,22H           ;ANFÜHRUNGSZEICHEN
7B4B  C9            0055        RET
                    0056  ;-------------------------------------------------
7B4C  205B5C5D      0057  TAB     DEFB    20H,5BH,5CH,5DH
7B50  5E5F2123      0058        DEFB    5EH,5FH,21H,23H
7B54  25274022      0059        DEFB    25H,27H,40H,22H
7B58  26243A3F      0060        DEFB    26H,24H,3AH,3FH
7B5C  28293E3C      0061        DEFB    28H,29H,3EH,3CH
7B60  3D2B2D2A      0062        DEFB    3DH,2BH,2DH,2AH
7B64  2F3B2C2E      0063        DEFB    2FH,3BH,2CH,2EH
7B68  30313233      0064        DEFB    30H,31H,32H,33H
7B6C  34353637      0065        DEFB    34H,35H,36H,37H
7B70  3839617B      0066        DEFB    38H,39H,61H,7BH
7B74  7C7D7E7F      0067        DEFB    7CH,7DH,7EH,7FH
                    0068  ;
                    0069  ;-------------------------------------------------
                    0070  ;-----  *****   LPRCHAR *****  -------------------
                    0071  ;-------------------------------------------------
7B78  E5            0072  LPRCHAR PUSH    HL
7B79  F5            0073        PUSH    AF
7B7A  21C229        0074        LD      HL,29C2H        ;HL:ADR. VON PORT C
7B7D  3E7F          0075  BUSY    LD      A,7FH           ;TEST AUF BREAK
7B7F  DBFE          0076        IN      A,0FEH          ;       "
7B81  1F            0077        RRA                     ;       "
7B82  300F          0078        JR NC,  BREAK           ;       "
7B84  7E            0079        LD      A,(HL)          ;TEST AUF BUSY:
7B85  1F            0080        RRA                     ;BIT 0 vom PORT C
7B86  38F5          0081        JR C,   BUSY
7B88  F1            0082        POP     AF              ;AUSGABE VON A
7B89  32C029        0083        LD      (29C0H),A       ;AN PORT A
7B8C  AF            0084        XOR     A               ;STROBE AUF NULL
7B8D  77            0085        LD      (HL),A          ;       "
7B8E  3E80          0086        LD      A,80H           ;STROBE AUF EINS
7B90  77            0087        LD      (HL),A          ;       "
7B91  E1            0088        POP     HL
7B92  C9            0089        RET
                    0090  ;-------------------------------------------------
7B93  CF            0091  BREAK   RST     8H              ;BREAK MIT
7B94  0C            0092        DEFB    0CH             ;ERR-CODE 0DH
                    0093  ;-------------------------------------------------
```

Bild 3. Druckerausgabe eines ZX-String in ASCII

Ein neues Konzept der Labordaten-Messung und Aufzeichnung

AD-5312
MULTILOGGER



- 24 Kanäle, ACV, DCV, OHM, Thermoelemente und Pt-100
- programmierter Scanner und Drucker/Plotter für Datenregistrierung
- integrierte Intelligenz, 5 Mathematik-Funktionen und Komparator

Siber Kikai K.K.
Zweigniederlassung Deutschland
Rossertstr. 2 · 6000 Frankfurt/M. 1
Telefon: 0611/717121
Telex: 411380 shf. d

Zu besichtigen auf der Hannover-Messe, Halle 13, Stand 911

Geppert, Kuhn, Marcus, Ney, Sickert

Automatische Spracheingabe und Sprachausgabe

Analyse, Synthese und Erkennung menschlicher Sprache mit digitalen Systemen.
432 Seiten, Hardcover, 275 Abbildungen, DM 73,—
Verlag MARKT & TECHNIK — Geschäftsbereich Fachliteratur — 8013 Haar bei München.
Erscheinungsjahr 1983.
ISBN 3-922120-31-8

Das Buch behandelt die Vorgänge beim Sprechen und Hören, die Grundlagen der digitalen Signalverarbeitung und die für die Analyse und Synthese von Sprachsignalen üblichen Methoden. Dieses neue und umfangreiche Gebiet wird zum ersten Mal geschlossen in deutscher Sprache dargestellt.

Das Buch setzt allerdings einige mathematische Grundkenntnisse voraus. Als Leserkreis dürften deshalb vor allem Ingenieure und Studenten in Betracht kommen. Wegen der umfassenden Darstellung eignet sich das Buch als Nachschlagewerk ebenso wie zum Selbststudium. Durch die zahlreichen Literaturhinweise können sich alle Interessierten über spezielle Fragen weitergehend informieren.

Die Grundbegriffe der digitalen Signalverarbeitung werden, vom Abtasttheorem bis zur Z-Transformation, knapp aber gründlich behandelt.

Die akustischen und linguistischen Grundlagen, insbesondere die Mechanismen des Hörens und Sprechens, stellen die Autoren in anschaulicher Form dar.

Die bei der Sprachanalyse und Sprachsynthese vorgestellten Methoden sind auf dem neuesten technischen Stand. Computergesteuerte Sprachsynthesizer sind seit einiger Zeit im Einsatz. Die automatische Spracherkennung dagegen ist wegen des komplexen Charakters der Sprache noch in den Anfängen, aber bereits bei weitem das umfangreichere Gebiet. Die Verfahren der Mustererkennung, die Sprach- und Sprechererkennung nehmen daher einen breiten Raum ein.

So bietet das Buch die Verknüpfung der akustischen Grundlagen mit der digitalen Signalverarbeitung. Es ist in einer klaren und treffenden Ausdrucksweise geschrieben. Die diversen Abbildungen und Diagramme verdeutlichen den Text und liefern einen wesentlichen Beitrag zum besseren Verständnis.

Detaillierte Produktübersichten und ca. 400 Literaturhinweise ergänzen die Darstellungen.

Alles in allem also ein erfreuliches und empfehlenswertes Buch. AT

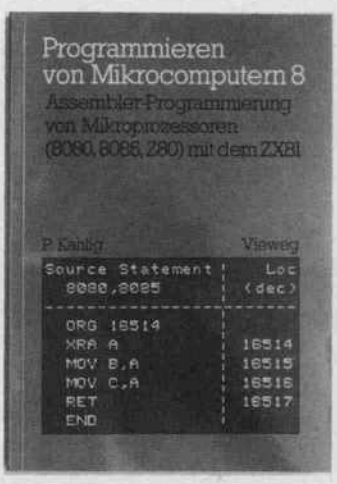
Peter Kahlig

Programmieren von Mikrocomputern Assembler-Programmierung von Mikroprozessoren (8080, 8085, Z 80) mit dem ZX 81

Braunschweig: Vieweg 1983
186 S., 100 Programmbeispiele
Kart. DM 38,—
ISBN 3-528-04238-9

Anwender von Mikrocomputern, die in der BASIC-Programmierung bereits erfahren sind, haben oftmals den berechtigten Wunsch, maschinennah, nämlich auf Assemblerebene, zu programmieren.

Das Buch ist, auch wenn der Titel es manchem flüchtigen Betrachter suggerieren mag, **kein** Assemblerlehrgang für ZX 81-Besitzer, die erste Gehversuche in Maschinensprache erwägen. Es ist vielmehr eine Sammlung einfacher 8-Bit-Arithmetikroutinen, also von Programmen für die vier Grundrechenarten (einschließlich der Kehrwertbildung) für natürliche Zahlen kleiner 256. Alle Assemblerroutinen werden parallel für die Mikroprozessoren 8080/8085 und Z 80 erstellt. Verkürzte Programmversionen für den Z 80, der den leistungsfähigeren Befehlssatz hat, sind, wo dies möglich ist, aufgeführt.



Besitzern des Kleincomputers ZX 81 ermöglichen die gelisteten BASIC-Programme, alle Arithmetikroutinen hexadezimal einzugeben, abarbeiten zu lassen und sich die Ergebnisse auf dem Bildschirm anzusehen.

Betrachtet man Programmierung als Umsetzung verschiedener Rechenvorschriften (Algorithmen) in eine für den Computer verständliche Form, so muß das Fehlen jeglicher Ablaufpläne bemängelt werden. Sind die den Arithmetikprogrammen zugrunde liegenden Algorithmen dem Leser nicht bekannt, so muß er sie sich mühsam aus den Assemblerlistings erarbeiten. Obwohl dies durch eine gute Kommentierung der Programme unterstützt wird, sollte der Leser zumindest die Grundrechenarten im Dualzahlensystem beherrschen. Zudem muß er die wichtigsten Assemblerbefehle und ihre Wirkungsweise kennen.

Um die Verschiedenartigkeit und Vielfältigkeit von Programmierlösungen aufzuzeigen, werden zahlreiche Beispiele aus der Sekundärliteratur modifiziert. Kahlig arbeitet jedoch die einzelnen Kriterien nicht in einer allgemein verständlichen Form heraus. Dem schon reichlich vorhandenen Programmangebot werden lediglich die geänderten Listings angefügt. Der schlechte Programmierstil einiger Programme, zum Beispiel die Verwendung spezieller Befehle entgegen ihrer eigentlichen Funktion, wird sogar als trickreich hervorgehoben.

Empfehlenswert nur für ZX 81-Besitzer mit beträchtlichen Vorkenntnissen. US

Dr. Peter Albrecht

Das Datenbanksystem dBaseII

Haar bei München, 1983
Verlag Markt und Technik
284 Seiten, kart. DM 68,—
ISBN 3-922120-36-9

Ein erfolgreiches Softwarepaket für die Datenverarbeitung wird in diesem Buch von einem Fachmann vorgestellt, der sich nicht nur mit Datenbanken auskennt, sondern auch in der Lage ist, sein Wissen an den Leser weiterzureichen. Nach einer knappen und präzisen Einführung in die Befehle von dBaseII kommt der Autor schnell zum Entwickeln von dBaseII-Programmen.

In kurzen, verständlichen Beispielen erklärt er Schleifen, Prozeduren und spezielle Programmiertricks wie zum Beispiel den Schutz eines Anwenderprogrammes durch Passwords und dialoggesteuertes Erzeugen von Programmen. Darüber hinaus zeigt er die Stärken (seltener die Schwächen) von dBase auf, besonders, wenn es um die Verknüpfung von Mikrocomputern mit großen Datenbanksystemen geht. Er beschreibt unterstützende Software und bietet eine knapp und gerade deshalb gut zu lesende Befehls-, Funktionenübersicht und einen wohlsortierten Index. Ein Buch, das sowohl als Einführung für den Newcomer als auch als Nachschlagewerk für den Datenverarbeiter sehr gut geeignet ist.

AB

Auch wir sind in

Hannover

c't magazin für computer technik

Besuchen Sie c't auf der Hannover-Messe 4. — 11. April 1984 Halle 12, II. OG, Stand 3200

Drucker

gibt es viele Typenrad- u. Nadeldrucker von: **Epson, Olivetti, Nec, Qume, Seikosha, Star, Brother, Juki, Mannesmann, Philips** ...

Doch

den Richtigen für SIE und Ihr Problem gibt es bei uns. Über Superpreise reden wir nicht. Wir haben Sie!

Computer

große und kleine, bekannte, weniger bekannte, ob **Televideo, Epson, Genie, Kaypro, Olivetti,** ob unser ASS **'Axio'**, ... Fragen **SIE** nach **Ihrem** Computer, er wartet darauf, Ihnen sein Können zu demonstrieren. Geballte Leistung zu unglaublichen Preisen.

Software

darüber reden viele, wir haben **die,** die Ihre Probleme löst.

Zubehör

Disketten-Kassetten-Bänder, Papier-Boxen-Farbbänder, **traumhafte Preise**

Wirklich nicht zu glauben, darum:

Anwender — Freaks — Händler

für jeden das RICHTIGE zu ..... Preisen??

Drucker **Computer** **Zubehör**

S.T.A. Data Control Corp.
Türmergasse 25 · 6900 Heidelberg
Telefon 0 62 21 - 78 05 55

STA

NEWMAN **Computer~ Versand**
Rolf W. Neumann

Alles für den Home-Computer

NEWMAN BERATUNGS-KATALOG

Gleich anfordern!
Kostenlos und unverbindlich erhalten Sie den großen Home-Computer Beratungs-Katalog mit vielen tollen Angeboten. Einfach Coupon einsenden.

GRATIS

Wir haben nur Qualitäts-Markenartikel zu Preisen, die uns so leicht keiner nachmacht.
Wir sind ein Versandhaus nur für Home-Computer, Hardware, Programme und Zubehör.
Wir beraten Sie neutral und unverbindlich. Am besten rufen Sie uns an. Wir helfen Ihnen weiter.
Wir liefern in der Regel innerhalb von 8 Tagen. Sofort-Lieferbestätigung bei telefonischer Anfrage.
Und außerdem erhalten Sie die Original-Hersteller-Garantie auf alle Artikel.
Teilzahlung ab sofort auch möglich.

Commodore 64
Newman liefert alles sofort, wer kann Ihnen das sonst noch bieten! Die gesamte Peripherie original von COMMODORE sofort ab Lager lieferbar. Und das zu Preisen, die uns so leicht keiner nachmacht. Fragen Sie unbedingt nach unseren Paket-Angeboten. **Sie werden staunen.**

Neu! Commodore SX 64
tragbarer "C 64" 64 K, 170 KB, – Floppy und Farb-Monitor zum Sonder-Preis von DM **2.948,–**
(kein Druckfehler)

Sharp MZ – 721
64 K, integrierter Cassetten-Recorder, 10 Spielprogramme gratis **nur** DM **939,–**
(da nur begrenzte Stückzahl sofort lieferbar, bitte noch heute bestellen).

Sharp MZ – 731
wie MZ – 721, zusätzlich mit integriertem Vierfarben-Drucker, 20 Programme am Lager. **nur** DM **1.239,–**

Sinclair ZX Spectrum, 16 KB, RAM DM **394,90**

Sinclair Spectrum, 48 KB, RAM nur noch DM **539,–**

Dragon 32 32 K-RAM Super-Graphic DM **795,–**

Seikosha GP 100 A, Matrix-Drucker, 50 Zeichen/sec. DM **675,–**

Sanyo, Daten-Monitor, 2112, gestochen scharf, grüne Anzeige DM **299,–**

Anzeige in orange-farben, 2212 DM **309,–**

Spectravideo, SV 318 DM **888,–**

Spectravideo, SV 328 DM **1.248,–**
(die gesamte Peripherie ist auch lieferbar).

Epson-Drucker
besonders preiswert, z. B. RX 80, nur DM **1.148,–**
Außerdem lieferbar: Texas Instruments, Brothers und jede Menge Fachbücher, Spiel- und Lern-Programme, Drucker, Laufwerke, Monitore und und und ...

Am besten gleich nachfragen: **040/830 26 27**
040/830 28 29

Ausschneiden auf Postkarte kleben (60 Pf Porto)

Ja, ☐ bitte senden Sie mir sofort kostenlos und unverbindlich Ihren Beratungs-Katalog.

Für Ihre Bestellung bitte hier eintragen. Alle Preise incl. MWSt. zuz. Versand-Kosten. Lieferung per Nachnahme, **Teilzahlung ab sofort möglich!**

Artikel	Stück	Preis

Name/Vorname
Straße
PLZ/Ort
Vorwahl/Telefon-Nr.
Unterschrift
Alter:

NEWMAN Computer~Versand
Rolf W. Neumann, Postfach 571261, 2000 Schenefeld.

****
Commodore CBM-64
****

FORTH 64
mit Grafik & Sound
Modul 198,- DM

hod 64©
das Grafikmodul 179,- DM

IEEE-488
CBM Schnittstelle 228,- DM

A-CALC
Electronic Spreadsheet
Disc (Dt. Anl.) 146,- DM

Light Pen	85,–
ROM Board	98,–
5fach Modul Steckplatz	198,–

BÜCHER , SPIELE .....

Info I/84 gegen 2,- Briefmarken

bst compulronic
Burgstraße 126a
6000 Frankfurt 60
Tel. 0611-453857

OPERATOR

OPERATOR, die neue universell einsetzbare Eingabeeinheit, ist speziell konzipiert worden, um die Flexibilität moderner Microsysteme zu unterstützen.

1. **Das Haupttastenfeld:** Tastenanzahl: 64 Tasten, „ASCII" – „deutsch", Zeichensatz, 4 Cursortasten, Autorepeat auf allen Tasten, durch Repeattaste beschleunigbar, Alphablock-Taste.
2. **Das Zehnerfeld:** Tastenanzahl: 18 Tasten Nummern 1 bis 9 sowie 0 00 . % – + : RETURN
3. **Das Funktionstastenfeld:** Tastenanzahl: 15 Tasten ungeshiftet, 15 Tasten geshiftet

Tasteninhalt: standardmäßig vorprogrammiert für bekannte Rechnersysteme oder auf Kundenwunsch, z. B. für Personalcomputer, Industriesteuerung, Textverarbeitung, (WORDSTAR etc.) usw.

Definieren Sie sich Ihre eigene Tastatur

Alle Tasten des Zehnerfeldes, die Funktionstasten u. die Cursortasten sind mit einem zusätzlichen EPROM 2, 4 od. 8 K selber umzudefinieren. Ausf. Unterlagen inkl. Schnittstellen: parallel + seriell, V24 od. TTL, Handshake serienmäßig

AFC computer GmbH, Salmstr., 20, 5000 Köln 91, Tel. 02 21/83 80 00, Tx. 8 873 254

ASC-COMPUTER-SHOP
HIRSCHGRABEN 9–11 · 5100 AACHEN
☎ 02 41/2 52 26

ASC-Comp. 48 K, APPLE-komp.	1150.–
dito, mit 10er-Tastatur	1298.–
dito, 64 K, Z-80-CPU + 6502	1390.–
dito, mit 10er-Tastatur	1548.–
MOTHERBOARD 48 K, bestückt	730.–
NETZTEIL 5 A	198.–
16-K-Karte, geprüft	145.–
LANGUAGE-Karte, geprüft	155.–
Z-80-CPU-Karte, geprüft	155.–
80-Zeichen-Karte, geprüft	228.–
PAL-Karte, geprüft	198.–
PRINTER-Karte m. Anschlußk.	198.–
CONTROLLER f. DISK-DRIVE	168.–
RS-232-SCHNITTSTELLE	268.–
HF-MODULATOR	39.90
DISK-DRIVE	798.–
JOY-STICKS, 4 Tasten	47.50
Leerplatinen, 16 K usw.	39.90
ALPHATRONIC PC	1495.–
CP80-Nadeldr., EPSON-komp.	998.–
JUKI-6100-Typenraddrucker	2350.–
MONITOR, 12", grün, entsp.	278.–
MONITOR, 12", orange, entsp.	298.–
RGB-MONITOR, 12"	998.–

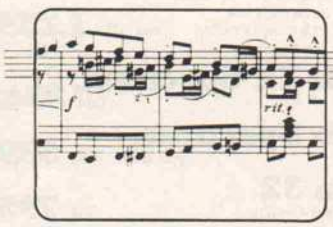
FORDERN SIE UNSERE PREISLISTE AN!

c't-Einzelheft-Bestellung

c't können Sie ab Ausgabe 12/83 direkt beim Verlag nachbestellen. Den Betrag von DM 6,— (zuzügl. DM 2,— für Porto und Verpackung) überweisen Sie bitte auf eines unserer Konten* oder fügen Sie Ihrer Bestellung einen Verrechnungsscheck bei.

Bei Bestellungen bis zu 4 Heften erhöhen sich die Kosten für Porto und Verpackung auf DM 3,—.

* s. Seite 16



Der INTERFACE AGE
Musik-Synthesizer
für den Commodore 64

EXTENDED SYNTHESIZER SYSTEM ist ein professionelles Musik-Synthesizer-System, das es Ihnen erlaubt, alle Sound-Möglichkeiten des Commodore 64 in vollem Umfang auszunutzen. Sie können fertige Musikstücke abspielen oder neue Kompositionen entwerfen. Die Noten samt aller Zusatzzeichen werden in grafisch hervorragender Weise in allen Details mit der Eingabe oder dem Spielablauf auf dem Bildschirm angezeigt. Alle Möglichkeiten, die es in der Musik gibt, bietet Ihnen EXTENDED SYNTHESIZER SYSTEM in drei Stimmen. Preis: DM 98,–!!!!

Auslieferung EXTENDED SYNTHESIZER SYSTEM für Commodore 64 und Floppy 1541 inklusive ausführlichem deutschen Handbuch mit ca. 50 Seiten.

BASIC-COMPILER

- Macht Ihre Programme wesentlich schneller.
- Kompatibel zu EXBASIC LEVEL II, SIMON'S BASIC und SOFTMODULEN.
- Ausführliche 40-seitige Dokumentation.
- Für Commodore 64/1541: DM 298,– inkl. MwSt.
- Für CBM 8032 mit 8050/8250: DM 698,– inkl. MwSt.

INTERFACE AGE Verlag GmbH
Vohburger Str. 1, 8000 München 21
Tel. (089) 5 80 67 02, Telex 5 213 489 iavmd

Ausführliche Gratisinformationen auf Anfrage.

Hannover Messe
Halle 3 CEBIT
U.S. Stand
47/48

10 Vorführgeräte BASF 6106
(3 Monate Garantie) ..... Stück **320,– DM**

3 Doppelgehäuse mit Netzteil
für BASF 6106 (Vorführgeräte
mit leichten Gehäusefehlern) ..... Stück **125,– DM**

EPROM-Programmiergerät

Programmiert: 2708, 2716, 2732/2532, 2764, 27128 u. ä., mit Zusatz auch 8748, 8749, 87555 usw.

Bausatz ..... **175,– DM**
Zusatz ..... **90,– DM**
Fertiggerät ..... **240,– DM**
Zusatz ..... **110,– DM**

Für Apple und kompatible Rechner

Floppy-Controler 5 ¼" für Industrielaufwerke, z. B. BASF 6106, Shugart usw. und Originallaufwerke

Bausatz ..... **195,– DM**
Fertiggerät ..... **280,– DM**

Preh-Commander-Keyboard

AK 87 mit Gehäuse, Anschlußkabel und separatem 10er Block, deutscher Tastensatz ..... **350,– DM**

Festplattenstation 10 MB brutto, mit sämtlicher Hard- und Software für Anschluß an Apple ..... **6000,– DM**

KÜHN ELEKTRONIK
2909 Bösel, Postfach 67, Telefon 0 44 94 / 15 64

Für TRS 80 ● Video-Genie ● Apple

Eprom-Programmiergerät SE 40
für 2716/2732/2532/2758, kompl. anschlußfertig,
Software auf Kass. oder Disk **Preis 269,— DM**

Eprom-Löschgerät SE 50
für max. 5 Eproms, Löschdauer ca. 10 Min.
Preis 125,— DM

afu electronic vertriebs gmbh
Steinstraße 9, 5778 Meschede, Telefon 02 91/75 85

DIE FORTH-QUELLE

- **CP/M 86** FORTH für **C't 86 Computer** ..... 376,20 DM
- Update für **CP/M 86 FORTH auf 32 Bit** Adressierung ..... 564,30 DM
- **Floatingpoint** Unterstützung ..... 376,20 DM
- **Textformatter + Editor** für CP/M 86 FORTH ..... 376,20 DM
- Alle Pakete auch für **Sirius** erhältlich

VC-20 FORTH Modul mit **ScreenEditor, Disk,** Utilities ..... 188,10 DM
Starting FORTH v. Leo Brodie ..... 55,60 DM
FORTH Encyclopedia, Beschreibung von FigFORTH ..... 88,20 DM
Source Listing für 1802, 6502, 6800 (6301), 8080, 8086, 9900, APPLE, 6809, NOVA, PDP 11, 68000, Z 80, Z 8000, Installationsmanual ..... je 40,00 DM

Wir versenden per Nachnahme (+7 DM) oder Vorauskasse (+3 DM).
Dies ist nur ein Teil unseres Programms. Infos bei:

DIE FORTH-QUELLE Angelika Flesch
Schützenstr. 3 · D-7820 Titisee-Neustadt · Tel. 8 76 51/16 65

SCANELEC-Sonderposten-Verkauf
Olympia Datensystem 6610

Buchhaltungscomputer mit Drucker (ohne Programm)
zum Umbauen — Ausschlachten — neu Programmieren

Vollelektronische Tastatur mit getrenntem 10er-Block und Sonderfunktionstasten, CPU-Platine mit 8080 A, 8255, 8257, 5101; div. EPROMs, 90% auf Sockel! Superstromversorgung.

Eingebauter 160 Z/sec Matrixdrucker (9 x 7 Matrix) mit 53 cm Druckbreite!! ..... **DM 600,—**

Dazu passend: Endlos-Formulartraktor mit eigenem Antrieb, 2 getrennte Formularantriebe (getrennt steuerbar!) auch für andere Geräte verwendbar ..... **DM 300,—**
Dazu passend: **Tischgestell mit Papierablage, sehr stabil** ..... **DM 200,—**
Paketpreis, frei nächster Bahnhof ..... **DM 999,—**

Alle Preise incl. MwSt., Lieferung per Nachnahme oder Verrechnungsscheck.

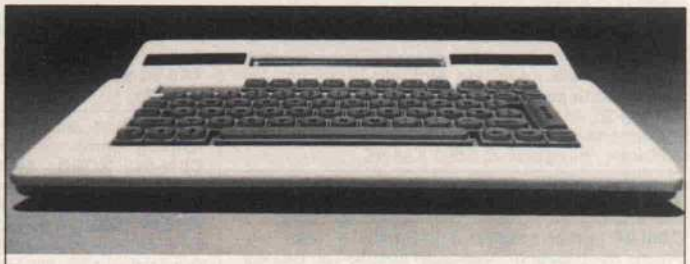
SCANELEC · Husumer Str. 1 · 2396 Sterup · 0 46 37/10 11

SHARP
computertechnik G. WEBER
EULENSPIEGELSTR. 56 * 8000 MÜNCHEN 83 * TEL: 089/601 2554

!!! absolute NIEDRIGPREISE !!!

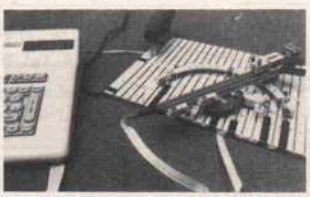
PC's aller Serien, Peripherie, Softwaremodule
für PC 1500 sowie 8,16,24 KB-Erweiterungen
MZ 3500 Serie
MZ 700 Serie
*Supergrafik
*Floppys
*anschlußfertige EPSON-Drucker

*EPSON**HEWLETT PACKARD*
*APPLE*COMMODORE*SEIKOSHA
*BINDER*etc.**ZUBEHÖR**



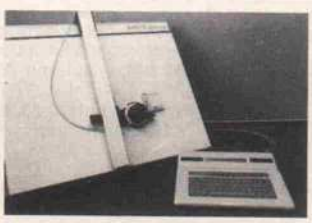
MAX 1

● Programmieren
● Editieren
● Schreiben



● Steuern
● Regeln
● System-Training

● Prüfen
● Testen
● Messen



● Vermessen
● Digitalisieren
● Positionieren

Dietmar Böhm · Elektronik
Schömberger Straße 4—6
7542 Schömberg-3
Telefon 0 70 84/76 00

in erweitertem **BASIC** mit **MAX 1**

FORTH — die neue und einfache Sprache: Assemblerlistings für alle wichtigen Mikroprozessoren DM 35, FORTH Installation Manual DM 35. Informationsmaterial kostenlos. C. Schmidt — Software, Bungestr. 8, 3500 Kassel.

SUCHE preiswerten **ZX81-Drucker.** Tel. 0658/77007.

CBM-64 immer neueste Top-Prgme ★ über 500 Prgme auch Beschr. Verkauf/Tausch. Tel. 06193/64456.

Fernschreiber mit Interf. für TRS80/Video-G., SC/MP-Mikrocomputer, je 180 DM, (089/8343864).

16 Bit-CPU und Peripherie 8086A 5MHz à 58,35, 8284 Clk-Gen à 14,30, 8288 BUSCTR à 39,50, 8L59 PIC à 14,30, R6545-1 à 39,50 DM. Alle Preise inkl. MwSt. Nachnahme-Versand. Fa. DGM ELECTRONIC, Gudrunstr. 16, 8000 München 19.

TRS80-Color-Computer Hard- u. Softw. 06175/1510.

CT 86 und IBM kompatible!!! Anwender für Programm- und Erfahrungsaustausch gesucht. Hermes-Gehrke, Tel. 02221/724636 oder 599 35 15.

CBM und **VC-20: Spitzensoftware** aus allen Bereichen in gr. Auswahl! Günstige Preise! Z.B. **Adress/Textv.** je 15 DM. **Maxiliste** 1 DM bei T. Hanne, Hertastr. 24, 4650 Gelsenkirchen.

Verkaufe **BBC-Computer/B** + Software + Interfaceschaltungen. Tel. 0651/76284.

Schaltbild für Oric + Spectrum 02303/13345.

Modulkopierer für VC-20 und C64, kopiert jedes Rom-Modul! Info 0,80 Rückporto. G. Genech, Derner Str. 363, 4600 Dortmund 14.

BÜRO-COMPUTER auf TRS-80/I-Basis (48k, 2 Disks MPI, Percom Doubler II) alles, incl. CPU + Interf. in Untertisch-Stahlgehäuse. Flache Tastatur und grüner Monitor (Epsonbeige). Kleinschr. + Umlaut auf Tast. u. Bschirm. Dazu ca. 50 Disks voll Software mit Dokumentations- + zusätzl. Literatur weg. Systemumstellung zu verk. DM 3500,— VB. Tel. 040/796 75-06(07) ab 19 Uhr.

CBM 64: 10 ausgesuchte Spitzenspiele 35 DM, einwandfrei auf Disk. o. Cass. Info gratis von: A. Nübel, Graefestr. 40, 1000 Berlin 61.

COMMODORE 64, alle Programme. Tel. 0201/602706.

64-FORTH. Nun gibt es diese schnelle strukturierte Sprache auch f. d. **C-64. FORTH** ist bel. erweiterbar u. damit für Graphik u. Soundprogr. gut geeignet. FORTH a. Cass. DM 49,—, Handbuch DM 25,—. INFO D. LUDA, Staudingerstr. 65, 8000 München 83 ★ 64-FORTH ★ 64-FORTH ★ 64-FORTH

FORTH FÜR APPLE!! und Verwandte mit Assembler und Editor **auf Cassette** LO-Graphik implementiert, HI-Graphik auf Source-Screens. Für DM 80,— bei J. Kaiser, Hausmattenweg 5, 7801 Oberried.

VERKAUFE Neu VC-20 + DATA SETTE + Programmierbuch, VB 380 DM. W. ZHANG, Zi 47, Tel. 0228/656961.

NEU! **MICRONIC**-Textsystem für **APPLE 2** nur 48 DM auf Minidisk. M. SCHWEIGER, PF 111069, 6300 Gießen.

TRS 80 MOD I Lev. II verk. W. Gutsch, Tel. 0228/613929.

CBM-64 Soft- und Hardware **ZX-SPECTRUM** Originalprogramme zu günstigen Preisen! Z.B.: für **CBM-64: KONG, MOON-BUGGY, BLUE-MOON, NEO, CLYPS** u.v.a. ab 29,– (Superhit: **ASSEMBLER-64:** 34,–) für **SPECTRUM: FORTH** 79,–, **PENGY, QUINCY, PANIC, ARCADIA** ab 19,90, Tastatur (Metallgehäuse) 169,–. Info: Rporto-Triebner, Bussardweg 39, 858 Bayreuth.

Cbm64–Cbm64–Cbm64: Achtung! Supersoftware aus allen Bereichen **(Spiel-User-Datei- und Textverarbeitungsprogramme** sowie **Musik-Basic- und Grafikerweiterungen).** Auch die neuesten Spiele, z.B. Congo Bongo, Pit Stop, Night Mission u.a., Gratis-Info sofort anfordern bei: Postlager-Nr. 012108B, 1000 Berlin 12.

COMPUTER-BÜCHER für Einsteiger, Fortgeschrittene, Profis! Große kommentierte Liste gratis: Versandbuchhandlung DUTSCHKE, Brüninghеide 97/CT, D-44 Münster, Tel. 0251/214459.

ACHTUNG APPLE-ANWENDER, Einführungsangebot: 2 TEAC-FLOPPYS FD55F mit Controler, Kabel, Gehäuse inkl. MwSt. und Montage DM 1995,–, Reparatur-Service aller gängigen Floppys, Drucker usw. Tel. 02107/60269.

CBM-64 90 Top-Programm auf Cass./Disk. DM 50,–, VS/Schein an M. Günsche, Odenwaldstr. 13, 6980 Wertheim 2, INFO gegen DM 2,– in Briefmarken. Lieferung innerhalb 3 Tagen.

SOFT/HARDWAREAUTOREN gesucht. Bieten gute Konditionen. Fordern Sie unsere Info-Mappe an. Oder senden Sie uns Ihr Programm mit kurzer Beschreibung. Computertyp egal. Antwort folgt sofort! BIMA-ELEKTRONIK, Heisterweg 6, 2382 Kropp.

Suche Computerfans in Berlin, die sich auch ihren CT-86 bauen, zum Erfahrungsaustausch. Günter Wegener, 1/30, Tel. 030/2163719.

NEC7220: Grafik-Software in Z80 Assembler. Listing (1700 Z.) oder 8" SSSD. 150,–. Tel. 0761/580795.

Verkaufe: AIM 65, 4K + 32K DRAM + Netzteil + Gehäuse, Fernschreiber (mit Interface + Softw.), Terminal ITT 3210, zus. IBM 2741 (Kugelkopf), auch einzeln. Rüdiger, Tel. 0228/465271.

68 000 FORCE PROFIKIT 1 mit 128 KB RAM und Schaltplan. FP 1500,–, **8"-DSDD** Laufwerke (2) unbenutzt. Fabr. MFE à 850,– zus. 1600,–. A. Pflughaupt, Nachtigallweg, 2351 Trappenkamp.

BASIS 108 kpZt. m. 2 SIEMENS DRIVES u. div. Software, DM 3850,–, SIEMENS DRIVE 7DD 100–5 neu, DM 465,–. SCHNITZER, Tel. 02932/7185.

KAYPRO II 8bit Rechner, 64KB RAM, Betr. Syst. CPM, 2 Floppy à 190KB, 9" Monitor, 10 Programmdisk., Origin. verpackt, 4700,– DM, Tel. 07475/1446

Neu auf 16-bit: **MULTIBASE**. Verbinden Sie dBASE II, MULTIPLAN u. Wordstar (bzw. andere TV-Programme) zu einem integrierten System. Infos über Knögel, Düsseld. Str. 7, 8000 München 40 oder Tel. 07940/2970.

Drucken mit Spectrum + VC-20/64! Grafikfähige Centronics-Schnittstelle (Software im Eprom) **nur DM 148,—** kpl. mit 80-Zeichen-Grafik-Drucker ab DM 675,—. Info „CT-1" anfordern. **DORSCH-electronic, Hauptstr. 23, 8501 Eckental 2, Tel. 09126/7419.** Händleranfr. willkommen.

KOMTEK I SUPER 32K zu verk. VB 700,– DM, W. Gutsch, Tel. 0228/613929.

Modul-Kopierer für VC-20 oder C-64. Kopiert jedes ROM-Modul problemlos. Netzunabhängig! Nur 98,–. Info gegen Rückporto. G. Genech, Derner Str. 363, D-4600 Dortmund 14.

EPROM-PROGRAMMIER-SERVICE 2716/2758 + 2732. Kopieren, löschen / Apple komp. Ursoft / Info 80 Pf., Klaus Becker, 6072 Dreieich, Eisenbahnstr. 83.

CP/M CROSS ASSEMBLER. Indeal für Einplatinencomputer Entwicklung 6809, 6800, 65XX-C02-II je DM 249,–, 8 Zoll sssd o. Apple CP/M format. D. Corson, Röntgenstr. 3, 59 Siegen, Tel. 0271/88815.

MATRIX-DRUCKER (Centronics 730) gebraucht, für **DM 389,–** von Privat. Tel. 040/632253.

SPECTRUM, SUPERSOFTWARE, SPECTRUM. Info 80 Pf., Dipl.-Ing. G. Verse, 4650 Gelsenkirchen, Grüner Weg 45.

COMMODORE 64-BESITZER! Für Software und Erfahrungsaustausch meldet Euch bei ERNST ZEDI, Dellsbergerallee 30, CH-**4053 BASEL.**

UMSTÄNDEHALBER: alphatronic-Pc (4 Wochen alt), incl. techn. Unterl. (Schaltpläne, HW-ADR.!), VB: DM 1400,–, Tel. 07231/79764.

Software C 64 zu kaufen und tauschen, Tel. 030/3346158.

STECKER für Computer liefern wir ab Lager sowie sämtliche Kabelverbindungen nach Ihren Angaben. Preisliste gegen 2,– DM in Briefmarken. COMPUTERSYSTEME NIEDERGESAESS, GOEBENSTR. 26, 6200 WIESBADEN, Tel. 06121/45919.

An dieser Stelle könnte Ihre private oder gewerbliche Kleinanzeige stehen. Exakt im gleichen Format: 8 Zeilen à 45 Anschläge einschl. Satzzeichen und Wortzwischenräumen. Als priv. Hobby-Elektroniker müßten Sie dann zwar 31,92 DM, als Gewerbetreibender 52,90 DM, Anzeigenkosten begleichen, doch dafür würde Ihr Angebot auch garantiert beachtet. Wie Sie sehen.

APPLE comp. Hdl-Liste, Tagestiefstpreise, Rückgaberecht 10 T. GENERALIMPORTEUR STREIL, Mommsenstr. 3, 4006 Erkrath 2, Tel. 02104/43079.

ZX Spektrum – ZX81 – VC20 Hardware, Zubehör im Selbstbau zu tollen Preisen. Sofort Info gegen Rückumschlag anfordern: BALTES, Nordring 60, 6620 Völklingen.

COMPUTER-CASSETTEN – 10er Pack, BASF-Band LHD mit Boxen, Etiketten und Einlegern, C 10 DM 15,–, C 20 DM 16,–, C 30 DM 17,–, C 40 DM 18,–. **CASSETTEN-AUFKLEBER** – 100 St. auf Lochstreifen 5,– DM, 120 St. auf A4-Druckbögen 7,– DM. **VIDEO-CASSETTEN** – Domiphon VHS E-180 17,80 DM, Christomenia-Cassettenstudio, Postf. 3584, Zwesten, 05626/281, Versand per Rechnung ab 20,– DM.

PC-1500 FORTH: Modulare Programmiersprache. CE155 erforderl. Cassette + Handbuch DM 129,–. INFO anfordern. A. Ulrich, Kopernikusstr. 11, 8000 München 80.

SPITZENSOFTWARE f. TI 99/4A. Nicht nur Invaderspielchen, sondern gute Software. Gratisliste TI99 bestellen!! G. Mahlig, Auf dem Heidenberg 19, 5000 Köln 60, Tel. 0221/7122358 PS. Suche TI Hardware.

Kurz + bündig.
Präzise + schnell.
Informativ + preiswert.

Wenn Sie Bauteile suchen, Fachliteratur anbieten oder Geräte tauschen, verkaufen oder kaufen wollen — mit wenigen Worten erreichen Sie durch c't schnell und preisgünstig jede Menge mögliche Interessenten.

Probieren Sie's aus! Die Bestellkarte für Ihre Kleinanzeige finden Sie am Schluß dieses Heftes.

Übrigens: Eine Zeile (= 45! Anschläge) kostet nur 3,99 DM. Inklusive Mehrwertsteuer!

Software: Branchenpaket „Reisebüro" — Textverarbeitung „Fibu" — u. a.
Hardware: Die professionelle Konfiguration zu Ihrem Problem
Service: Auslieferung/Wartung max. 24 Std. bundesweit
Preis: Ob Händler oder Enduser, rufen Sie uns an
Türmergasse 25 · 6900 Heidelberg · 0 62 21/78 05 55

STA

c't magazin für computer technik

6/84 — Anzeigenschluß am 11. April 1984

SIDNEY II

SIDNEY - die Computer für den professionellen Anwender!

ausgestattet mit 2 Prozessoren
SIDNEY II: 6502 und Z-80
SIDNEY 88: 6502 und 8088
64 K RAM aufrüstbar bis zu 512 K
incl. PROFI-Tastatur ab DM 3640,00

ALLES für den APPLE II

PROFI-Tastatur, einzeln	DM	698,00
Hard Disk 10 MB, kompl. anschlußf., im Gehäuse	DM	5998,00
5" Flachlaufwerk anschlußfertig	DM	589,00
Plot-II Graphics Table	DM	269,00
SUPER JOYSTICK	DM	118,00
Minidisks, "CONTROL DATA", SS/DD 10 St. ab	DM	55,00
Diskettenbox für 80 Mini-Disks, mit Schloß	DM	59,00

Alle Preise incl. Mehrwertsteuer
Gratispreisliste anfordern!

Wiederverkäufer-Preise nur auf schriftliche Anfrage.

HIB GMBH
MICROCOMPUTER
Wunderburggasse 2
8500 Nürnberg 1
0911 / 22 13 83

TRACE 80 – ICE

TRACE 80 – kompatibler Slave-Emulator für CP/M-Computer

- Anschluß für V24
- Echtzeit-Emulation bis 6 MHz ohne Wait
- Beliebig viele Haltepunkte, auch im ROM
- 64-K-Emulations-RAM
- Über 50 Befehle
- Symbolisches Debugging mit M 80, RMAC oder MS-Compilern
- Strukturiertes Debugging
- Ablaufsteuerformeln
- Erweiterbar durch Trigger-/Trace-Logik

Lothar Lauterbach
Ing.-Büro für Datentechnik
Sommerstraße 10
8012 Riemerling
Telefon 0 89 / 60 61 73

c't-Platinen

c't-Platinen bestehen aus Epoxid-Glashartgewebe, sind fertig gebohrt und mit Lötstopplack versehen bzw. verzinnt. Die Bestellnummer bezieht sich auf den Beitrag, in dem das betreffende c't-Projekt vorgestellt wurde. Sie setzt sich zusammen aus Jahrgang, Heftnummer und Seitennummer. Die zusätzlichen Buchstaben bedeuten: 'd' — doppelseitig, 'B' — Bestückungsaufdruck, 'E' — elektronisch geprüft.

Nr.	Projekt	Format	Preis
831241dBE	Terminal A (ohne Tastatur)	ca. 84 x 234 mm	59 DM
831242dBE	Terminal B (mit Tastatur)	Doppel-Europa	75 DM
831262	Universelles Netzteil	Europa	14 DM
840147dBE	c't 86, CPU-Karte	Europa	85 DM
840148dBE	c't 86, RAM-Karte 256 KByte	Europa	88 DM
840149dBE	c't 86, I/O-Karte	Europa	69 DM
840288dBE	c't 86, Floppy-Interface	Europa	65 DM
840150d	Busplatine (96pol., 10 Steckplätze)	84 x 208 mm	49 DM
840184d	CEPAC-80 mit Wrap-Feld	Europa	69 DM
840187d	CEPAC-80 ohne Wrap-Feld	ca. 86 x 100 mm	49 DM
840242B	Centronics/V24-Interface für Olympia COMPACT	80 x 136 mm	15 DM
840252B	c't-Sprachsynthesizer	100 x 117 mm	21 DM
840352dB	CEPAC-65, Version A	80 x 100 mm	27 DM
840354dB	CEPAC-65, Version B	Europa	52 DM

So können Sie bestellen: Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck oder einen von Ihrer Bank quittierten Einzahlungsbeleg über die Bestellsumme zuzüglich 3 DM (für Porto und Verpackung) bei. Bei Bestellung aus dem Ausland muß stets eine Überweisung in DM erfolgen. Die Überweisung und Ihre Bestellung richten Sie bitte an:

c't-Versand, Verlag Heinz Heise GmbH, Bissendorfer Straße 8, 3000 Hannover 61, Konto-Nr. 9305-308, Postscheckamt Hannover

Firmenverzeichnis zum Anzeigenteil

adcomp Datensysteme, München .. 77
AFC Computer, Köln 104
afu, Meschede 105
ancon Computertechnik, Berlin 14
ASC, Aachen 104
Aumann, Strassenhaus 97

Behn, Düsseldorf 18
BICC VERO, Bremen 33
Böhm, Schömberg 105
BSP Krug, Regensburg 84
bst computronic, Frankfurt 104

CE Computer, Schwerte 14
COMPUSHACK, Neuwied 19
COMPUTER ACCESSOIRES INT'L, Ottobrunn 10, 11
Conitec, Darmstadt 97
Conrad, Hirschau 18
Cromemco, Eschborn 15

Data Becker, Düsseldorf 5

Elektronikladen, Detmold 18
eps, Großrinderfeld 31
EVG Electronic, Langenhagen 9

Forth Quelle, Flesch, Titisee 105
Frech-Verlag, Stuttgart 99
Frölje, Oldenburg 74

Glisch, Leonberg 18

HIB, Nürnberg 107
H + I Informatik, FL-Vaduz 35
Hösch, Düsseldorf 40

IMETEC EDV, Wiesbaden 99
Interface AGE-Verlag, München ... 104

Köller, Schieder-Schwalenberg 99
Kühn, Bösel 104

LAMPSON & ZERBE, Büttelborn ... 95
Lauterbach, Riemerling 107
LECH-TECHNICS, Kerpen-Türnich . 41
Linde, Schwäbisch-Hall 40
LOGITEK, Berlin 99

Magnetronic, München 18
MARFLOW, Hannover 87, 89
MC-SERVICE, Bielstein 18
Meinhold, Siegen 84
Meyer + Hanelik, Hannover 17
mirwald, Unterhaching 97
Münzenloher, Holzkirchen 35

NEC-Haupthändler 22
Neuhaus, Hamburg 41
NEWMAN Computer-Versand, Schenefeld 103

Preh, Bad Neustadt 112
profisoft, Osnabrück 65

RATEV, Ratingen 18
Röckrath, Aachen 18

SCANELEC, Sterup 105
Siber Kikai K.K., Frankfurt 101
SYNELEC, München 13
Syntax, Rastatt 99
Syscom, Winkler, Heidelberg 20
Systech, Braunschweig 91

Schanuel, Darmstadt 71

STA, Heidelberg 103, 107
Stede, Willingen 97
Strie, Schwafördern 67

Teepe, Weilrod 67
Time Soft, Stuttgart 35

üding, Menden 97
Unitronic, Düsseldorf 20

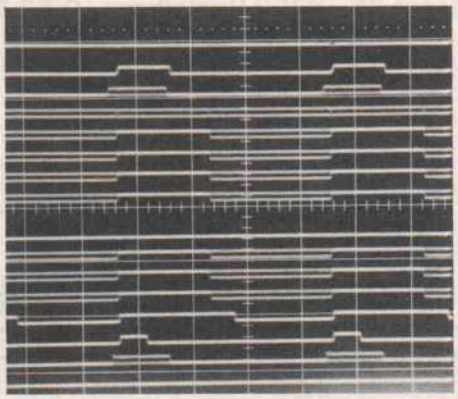
Weber, München 105
Wiesemann, Wuppertal 74
Wilke, Aachen 41

In der nächsten c't

unter anderem

Fehlersuche in Computer-Systemen

Professionelle Testgeräte für Mikrocomputer-Systeme sind für viele Selbstbauer unerschwinglich. c't zeigt, wie man sich behelfen kann, wenn lediglich ein einfaches Oszilloskop zur Verfügung steht. Das Projekt des Monats paßt in diesen Zusammenhang: Ein Vorsatzgerät, das es ermöglicht, 16 Kanäle gleichzeitig auf dem Oszilloskop-Bildschirm darzustellen. Der 'digitale Scope-Extender' verarbeitet TTL-Signale bis zu 20 MHz und kostet nicht die Welt (Bauteile und Platine: etwa 60 Mark).



Umgang mit MACRO-80

Microsoft hat mit dem MACRO-80 einen Assembler geschaffen, der (kaum) Wünsche offenläßt. Aber gerade die vielen Möglichkeiten, die das Programm bietet, machen dem Einsteiger das Leben schwer. Der c't-Bericht soll nicht das Handbuch ersetzen, sondern anhand von Demonstrationsprogrammen die wichtigsten Features verständlich machen und so ein 'Gefühl' für den Umgang mit diesem ausgezeichneten Werkzeug vermitteln.

Centronics-Interface für ZX Spectrum

Nach den ZX81-Besitzern erhalten nun auch die Spectrum-Anwender eine preisgünstige Möglichkeit, 'richtige' Drucker zu betreiben. Die c't-Schaltung läßt sich — wie beim ZX81 — nicht nur als Drucker-Interface, sondern auch als universelle Schnittstelle für die Ein-/Ausgabe einsetzen.

ORIC-ROM geknackt

Mit Informationen über Internes ist das ORIC-Handbuch sehr zurückhaltend. Viele Anwender beklagen, daß sie die Leistungsfähigkeit ihres Computers nicht voll nutzen können. c't beendet die Geheimniskrämerei: In der nächsten Ausgabe finden Sie alle wichtigen ROM-Adressen und weitere wertvolle Praxistips für den Umgang mit dem ORIC-1.

SuperTape für 6502-Computer

Schnelle und sichere Datenaufzeichnung auf der Kassette, Kommunikation zwischen Rechnern verschiedenen Fabrikats — all das ermöglicht das in dieser c't-Ausgabe vorgestellte SuperTape-Verfahren. Im nächsten Monat veröffentlichen wir die SuperTape-Routinen für 6502-Systeme am Beispiel der beiden populären Commodore-Homecomputer VC-20 und C 64.

Prüfstand:

Tandy Colour Computer 2

Erweiterte Möglichkeiten, mehr 'Benutzerfreundlichkeit' verspricht Tandy mit dem Nachfolger des Colour 1. Wird sich der Neue auf dem heiß umkämpften Markt der kleinen Homecomputer durchsetzen können?

Außerdem:

Mehr BASIC intern
VIA 6522-Applikation
SuperTape-Bedienungsprogramm für ZX81
u.v.a.m.

Heft 5/84 (April/Mai) erscheint am 12. April 1984
Änderungen vorbehalten

Das bringt elrad

elrad 3/84 — jetzt am Kiosk

● Die große Bauanleitung: ElMix — Mischpult für Bühne und Studio; mit einem 72 cm (!) langen Platinen-Layout im Maßstab 1:1 ● Bauanleitung Spezial-Hifi-Netzteil ● Bauanleitung 'Heizungsregelung für Minimalverbrauch' ● Die elrad-Laborblätter: Analog-Schalter in CMOS; über 20 Schaltungsbeispiele ● Computing-Today: Bauanleitung ZX81-Soundboard ● Sprachkurs 'Going FORTH' (Schluß)

elrad 4/84 — ab 27. 3. 1984 am Kiosk

● Bauanleitung Bühne/Studio: PA-Box ● Bauanleitung 'Elektronische Sicherung' ● Die elrad-Laborblätter: Der Operationsverstärker in Meß- und Prüfschaltungen; zahlreiche Schaltungsbeispiele ● elrad-Report: Musikmesse Frankfurt — die neuen Techniken ● Computing Today: FORTH-Simulator in ZX-BASIC

Impressum:

c't
Magazin für Computertechnik

Verlag Heinz Heise GmbH

Bissendorfer Straße 8, 3000 Hannover 61
Postanschrift: Postfach 27 46
3000 Hannover 1
Ruf (0511) 53520

technische Anfragen nur freitags 9.00—15.00 Uhr

Postscheckamt Hannover, Konto-Nr. 9305-308
Kreissparkasse Hannover, Konto-Nr. 000-019968
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Andreas Burgwitz (stellvertretender Chefredakteur)
Dipl.-Ing. Detlef Grell

Ständige Mitarbeiter:
Dipl.-Ing. Rolf Keller
Dipl.-Ing. Eberhard Meyer
Dipl.-Chem. Holger Petersen
Dipl.-Ing. Eckart Steffens
Dipl.-Ing. Kurt Werner

Technische Assistenz: Hans-Jürgen Berndt

Abonnementsverwaltung, Bestellwesen:
Dörte Imken, Anita Kreutzer

Anzeigen:
Wolfgang Penseler (Anzeigenleiter)
Gerlinde Donner (Disposition)

Es gilt die Anzeigenpreisliste 1 vom 1. 10. 1983

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:
Verlag Heinz Heise GmbH
Postfach 27 46
3000 Hannover 1
Ruf (0511) 53520

Herstellung: Wolfgang Ulber

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Satz und Druck:
Hahn-Druckerei, Im Moore 17, 3000 Hannover 1
Ruf (0511) 708370

c't erscheint monatlich.
Einzelpreis DM 5,—, öS 52,—, sfr 6,—, hfl 6,80,

Jahresabonnement Inland DM 58,— inkl. MwSt. und Versandkosten. Schweiz sfr 58,— inkl. Versandkosten. Österreich öS 480,— inkl. Versandkosten. Niederlande hfl 68,— inkl. Versandkosten. Sonstige Länder 65,— DM inkl. Versandkosten.

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):
Verlagsunion Zeitschriften-Vertrieb
Postfach 5707
D-6200 Wiesbaden
Ruf (06121) 266-0

Verantwortlich:
Textteil: Christian Persson
Anzeigenteil: Wolfgang Penseler
beide Hannover, Bissendorfer Straße 8,
3000 Hannover 61

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Die geltenden gesetzlichen und postalischen Bestimmungen bei Erwerb, Errichtung und Inbetriebnahme von Sende- und Empfangseinrichtungen sind zu beachten.

Die gewerbliche Nutzung, insbesondere der Schaltpläne und gedruckten Schaltungen, ist nur mit schriftlicher Genehmigung des Herausgebers zulässig. Die Zustimmung kann an Bedingungen geknüpft sein.

Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Nachdruck nur mit Genehmigung des Verlages. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das Exclusivrecht zur Veröffentlichung. Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden.

Sämtliche Veröffentlichungen in **c't** erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Printed in Germany
© Copyright 1984 by Verlag Heinz Heise GmbH

ISSN 0724-8679

Titelidee: **c't**
Titelfoto:
Fotozentrum Hannover, Manfred Zimmermann

c't - Private Kleinanzeige

Auftragskarte

Private Kleinanzeigen je Druckzeile DM 3,99 inkl. MwSt.

Gewerbliche Kleinanzeigen je Druckzeile DM 6,61 inkl. MwSt.

Chiffregebühr DM 5,70 inkl. MwSt.

c't - Kleinanzeige Auftragskarte

Bitte veröffentlichen Sie in der nächsten erreichbaren Ausgabe nachstehenden Text:

DM 3,99	
DM 7,98	
DM 11,97	
DM 15,96	
DM 19,93	
DM 23,94	
DM 27,93	
DM 31,92	

Pro Zeile bitte jeweils 45 Buchstaben einschl. Satzzeichen und Wortzwischenräume. Wörter, die **fettgedruckt** erscheinen sollen, unterstreichen Sie bitte. Den genauen Preis inklusive Mehrwertsteuer können Sie so selbst ablesen. Soll die Anzeige unter einer Chiffre-Nummer laufen, so erhöht sich der Endpreis um DM 5,70 Chiffre-Gebühr inkl. MwSt.

Bitte umstehend Absender nicht vergessen!

c't-Kontaktkarte

Mit dieser Service-Karte können Sie

- **Informationen** zu in **c't** besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierenden oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Folien, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, **c't**-Versand, Postfach 2746, 3000 Hannover 1, **ordern**.

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in **c't** _____/8__, Seite _____ erschienene

☐ Anzeige ☐ redaktionelle Besprechung

☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____

☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't-Platinen-Folien-Abonnement

Abrufkarte

Saubere Platinen stellen Sie mit der **c't**-Klarsichtfolie her. Sie ist zum direkten Kopieren auf Platinen-Basismaterial im Positiv-Verfahren geeignet.

Einzelbestellungen siehe Anzeigenteil.

c't-Platinen-Folien-Abonnement Abrufkarte

Ja, übersenden Sie mir für 1 Jahr die **c't**-Platinen-Folie ab

Monat _____ 1983

Das Platinen-Folien-Abonnement gilt nur für 12 Monate und muß im voraus bezahlt werden. Es kostet DM 30,— inkl. Versandkosten und MwSt.

☐ Postscheck Hannover, Konto-Nr. 9305-308;

☐ Kreissparkasse Hannover, Konto-Nr. 000-0199 68.

Bitte geben Sie unbedingt auf dem Überweisungsbeleg „Folien-Abonnement c't" an.

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Straße/Nr.

PLZ Wohnort

Datum/Unterschrift

Ich bestätige ausdrücklich, vom Recht des schriftlichen Widerrufs innerhalb von 10 Tagen nach Abschluß beim Verlag Heinz Heise GmbH, Postfach 2746, 3000 Hannover 1, Kenntnis genommen zu haben.

Unterschrift _____

Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

Absender (Bitte deutlich schreiben!)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Bitte veröffentlichen Sie den umstehenden Text von ____ Zeilen zum Gesamtpreis von _____ DM in der nächsterreichbaren Ausgabe von **c't**. Den Betrag habe ich auf Ihr Konto

Postscheck Hannover,
Konto-Nr. 93 05-308;
Kreissparkasse Hannover,
Konto-Nr. 000-0 199 68

überwiesen/Scheck liegt bei.

Veröffentlichungen nur gegen Vorauskasse.

Datum Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

Antwort

Bitte mit der jeweils gültigen Postkartengebühr freimachen

magazin für computer technik

Anzeigenabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

c't - Private Kleinanzeige

Auftragskarte

Nutzen Sie diese Karte, wenn Sie etwas suchen oder anzubieten haben!

Abgesandt am

_____ 198__

Bemerkungen

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen. ▶

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

_____ 198__

an Firma _____

Bestellt/angefordert

c't-Leser-Service

Antwort

Bitte mit der jeweils gültigen Postkartengebühr freimachen

magazin für computer technik

Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

c't-Platinen-Folien-Abonnement

Abrufkarte

Abgesandt am

_____ 198__

zur Lieferung ab

Heft _____ 198__

Jahresbezug DM 30,—
inkl. Versandkosten und MwSt.

Abbuchungen sind aus organisatorischen Gründen nicht möglich.

Ausführlich und umfangreich wie kein Lautsprecher-Selbstbau-Heft zuvor.

Spitzenqualität im Selbstbau! Von der Mini-Box bis zum Lautsprecher-schrank

Laser in der HiFi-Technik

Elektronik für Lautsprecherboxen
Passive Lautsprecherweichen
Lautsprecherweichen
Elektronische Frequenzweiche
Lautsprecherschutzschaltung

Bauanleitungen
Transmissionline-Lautsprecher
+ Nachlese
Corner Speaker
KEF CS5
E80-Magnat
Wharfedale E90
Transmissionline
Harbeth 250
+ Nachlese
+ Ergänzungen
FOCAL 250DB
Ton-Pyramiden
Donar —
der Subwoofer
Podszus-Baßhorn
Dynaudio Pyramide

Für die Bühne:
elrad-Jumbo

Versand-bedingungen:

Die Lieferung des Heftes erfolgt per Nachnahme (+ DM 4,50 Versandkosten) oder gegen Verrechnungs-scheck (+ DM 2,00 Versandkosten).

elrad EXTRA

14,80

Sie erhalten das Selbstbau-Heft direkt beim Verlag Heinz Heise GmbH, Postfach 27 46, 3000 Hannover 1

YOUR COMMAND BRIDGE.
PREH COMMANDER ELECTRONIC.

Die Keyboards mit dem besonderen Touch

- Programmierbares, intelligentes Keyboard
- mit Mikroprozessor
- für hohe Ansprüche

- Schnittstellen-Variationen
- Gehäuse im Preh-Design

- Überall dort, wo die Brücke zwischen Mensch und Maschine hergestellt werden muß, löst Preh die daraus resultierenden Aufgaben – und darin haben wir seit über 60 Jahren Erfahrung.

Preh Commander mit Encoder-Elektronik

Hannover-Messe '84
Halle 12, 2. OG. Stand Nr. 2262

Prädikat: if 83 Die gute Industrieform

Preh

Preh Elektrofeinmechanische Werke
Postfach 1740
D-8740 Bad Neustadt
Telex: 672503 · Tel. (09771) 921